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

Giró Moliner, Arnau. *Desenvolupament d'un joc d'escacs 3D amb control per veu i assistència per a persones amb discapacitat visual o de mobilitat mitjançant OpenGL*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334418>

under the terms of the  license.

Desenvolupament d'un joc d'escacs 3D amb control per veu i assistència per a persones amb discapacitat visual o de mobilitat mitjançant OpenGL

Arnau Giró Moliner

Resum—Aquest Treball de Fi de Grau presenta el desenvolupament d'un joc d'escacs en un entorn tridimensional mitjançant OpenGL, pensat per oferir una experiència d'oci plenament accessible a persones amb discapacitat visual o de mobilitat reduïda. El mecanisme principal d'interacció no és gràfic sinó vocal: el sistema incorpora reconeixement de veu (Vosk) per seleccionar i moure les peces, incloent-hi moviments especials com l'enroc i la promoció, i un sistema de síntesi de veu (Piper) que informa contínuament l'usuari de l'estat de la partida. El projecte inclou també un mode de joc en xarxa amb arquitectura client-servidor basada en un servidor extern intermedi i sockets TCP, que permet disputar partides entre dos ordinadors sense que cap jugador hagi d'actuar com a servidor exposat. El desenvolupament s'ha realitzat en C++ sobre Visual Studio, seguint una metodologia iterativa incremental amb control de versions a GitHub. S'han assolit els objectius d'alta i mitjana prioritat lògica completa del joc, control per veu, assistència per àudio i joc en xarxa, mentre que la integració d'un motor d'intel·ligència artificial (Stockfish) resta com a línia de treball futur.

Paraules clau—Accessibilitat, OpenGL, escacs 3D, reconeixement de veu, síntesi de veu, joc en xarxa, C++, Vosk, Piper.

Abstract—This Bachelor's Thesis presents the development of a three-dimensional chess game built with OpenGL, designed to provide a fully accessible leisure experience for people with visual or mobility impairments. The main interaction mechanism is not graphical but vocal: the system uses speech recognition (Vosk) to select and move pieces, including special moves such as castling and pawn promotion, and a speech synthesis system (Piper) that continuously informs the user of the game state. The project also includes a network multiplayer mode with a client-server architecture based on an external relay server and TCP sockets, allowing games to be played between two computers without either player acting as an exposed server. Development was carried out in C++ using Visual Studio, following an iterative-incremental methodology with version control on GitHub. The high- and medium-priority objectives were achieved: complete game logic, voice control, audio assistance and network play, while the integration of an artificial intelligence engine (Stockfish) remains as future work.

Index Terms— Accessibility, OpenGL, 3D chess, speech recognition, speech synthesis, network gaming, C++, Vosk, Piper.



1 INTRODUCCIÓ

Els escacs constitueixen un dels jocs de taula més antics i populars del món, i la seva adaptació a entorns digitals n'ha ampliat l'abast i l'accessibilitat. Tanmateix, la majoria de les implementacions actuals depenen de dispositius d'entrada tradicionals: ratolí, teclat o pantalla tàctil, fet que dificulta o impedeix l'accés a persones amb discapacitat visual o de mobilitat reduïda.

Aquest Treball de Fi de Grau planteja el desenvolupament d'un joc d'escacs en un entorn tridimensional mitjançant OpenGL, en el qual el mecanisme principal d'interacció no és gràfic sinó vocal. El sistema incorpora reconeixement de veu per a la selecció i el moviment de les peces, síntesi de veu per informar l'usuari de l'estat de la partida en tot moment, i un mode de joc en xarxa que permet disputar partides entre dos ordinadors diferents. D'aquesta manera, el projecte combina un repte tècnic, la programació gràfica 3D i el processament del llenguatge parlat amb un objectiu d'impacte social: oferir una eina d'oci plenament accessible

• E-mail de contacte: [1529328@uab.cat]

• Menció realitzada: [Computació]

• Treball tutoritzat per: Javier Sanchez Pujadas

• Curs 2025/26

per a persones amb discapacitat visual o de mobilitat.

El document s'organitza de la manera següent: la secció 2 revisa l'estat de l'art dels jocs d'escacs digitals; la secció 3 descriu la metodologia i les tecnologies de desenvolupament emprades; la secció 4 detalla la implementació del sistema; la secció 5 analitza els resultats obtinguts i el grau d'assoliment dels objectius; i la secció 6 recull les conclusions i les línies de treball futur.

Des d'un punt de vista més ampli, l'accessibilitat en l'àmbit del programari d'oci continua sent, sovint, una consideració secundària respecte al disseny gràfic i a la jugabilitat. En el cas dels escacs digitals, això es tradueix en interfícies pensades gairebé exclusivament per a una interacció visual i manual, fet que limita l'accés d'un col·lectiu d'usuaris per al qual aquest joc per la seva naturalesa estratègica i no dependent de la rapidesa física podria resultar especialment adequat. Aquest treball pretén, precisament, situar la veu com a primer mecanisme d'interacció en lloc de tractar-la com una funcionalitat afegida, integrant-la des del disseny inicial del sistema en comptes d'adaptar-la a posteriori sobre una interfície gràfica preexistent.

1.1 Objectius

L'objectiu general del treball és el desenvolupament d'un joc d'escacs en entorn 3D mitjançant OpenGL que incorpori mecanismes d'accessibilitat com el control per veu, així com funcionalitats avançades com el joc en xarxa i la integració d'un motor d'intel·ligència artificial. El sistema havia de garantir una experiència de joc funcional, visualment clara i progressivament ampliable, prioritzant primer la base estructural del joc i, posteriorment, les funcionalitats avançades.

Els objectius específics es van establir segons la seva criticitat i ordre d'implementació, organitzats en tres nivells de prioritat:

- **1.1.1 Nucli funcional (alta prioritat)**

- **Implementar la lògica completa del joc d'escacs:** moviment correcte de totes

les peces segons el reglament oficial, validació de moviments legals i gestió de situacions especials (escac, escac i mat, enroc, promoció).

- **Desenvolupar la representació gràfica 3D** del tauler i les peces amb OpenGL, garantint una correcta visualització espacial, diferenciació clara de peces i colors, i fluïdesa en la interacció visual.
- **Dissenyar una interfície bàsica** que permeti seleccionar i moure peces de manera intuïtiva, mostrar informació rellevant de la partida i jugar localment entre dos jugadors al mateix ordinador.

- **1.1.2 Prioritat mitjana**

- **Implementar un sistema de control per veu** que permeti introduir moviments mitjançant ordres parlades, interpretar coordenades i accions del joc, i facilitar-ne l'ús a persones amb discapacitat visual o de mobilitat.
- **Desenvolupar un sistema de joc en xarxa** amb arquitectura client-servidor que permeti la connexió entre ordinadors, la sincronització en temps real de la partida i una gestió bàsica de la comunicació entre jugadors.

- **1.1.3 Prioritat baixa**

- **Integrar un motor d'intel·ligència artificial** que permeti jugar contra la màquina mitjançant un algorisme de cerca (p. ex. Minimax amb poda alfa-beta), una funció d'avaluació del tauler i, si el temps ho permet, diversos nivells de dificultat.
- **Millorar l'estètica i l'experiència d'usuari:** optimització gràfica, millores visuals en peces i tauler, poliment de la interfície i correcció d'errors de rendiment.

2 ESTAT DE L'ART

En l'àmbit dels jocs d'escacs digitals existeixen diverses opcions consolidades [13], [14], des de jocs clàssics fins a plataformes modernes amb intel·ligència artificial i multijugador en línia. La taula I en resumeix les característiques principals.

TAULA I COMPARATIVA D'APLICACIONS D'ESCACS EXISTENTS

Aplicació	Interfície	Control	Multijugador
Chessmaster	2D/3D, tutorials i IA	Teclat / mando	LAN / Internet (servei propi)
Chess Titans	Tauler 3D detallat	Ratolí	Local (mateix PC)
Chess.com / ChessKid	Web i mòbil, 2D/3D	Ratolí / tàctil	Internet, temps real i torneigs
Lucas Chess	PC, 2D/3D personalitzable	Ratolí	No (local / IA)

La majoria d'aquestes aplicacions centren la interfície en el tauler, amb peces clarament codificades per color i taulers opcionals en 2D o 3D, atractius visualment però amb un cost de recursos més elevat en el cas dels gràfics 3D. La interacció es realitza pràcticament sempre per clic amb ratolí (PC) o tocar i arrossegar (mòbil); aplicacions com Chess.com o ChessKid optimitzen, a més, la interfície per a usuaris principiants amb accions clares com desfer moviments o suggeriments de la IA.

Pel que fa a la simplicitat i l'accessibilitat, el disseny general busca que l'usuari es pugui centrar en el joc sense distraccions, amb menús (nivell d'IA, tipus de tauler, multijugador) ben organitzats.

Cap d'aquestes solucions ofereix, però, una interacció per veu integrada de manera nativa: el control continua basant-se en dispositius d'entrada manual, cosa que deixa fora els usuaris amb discapacitat visual o de mobilitat reduïda. Aquest buit constitueix precisament l'espai d'innovació que aborda aquest projecte, en el qual la veu deixa de ser una funcionalitat accessòria per convertir-se en el mecanisme principal d'interacció amb el joc.

Aquesta mancança és coherent amb una tendència més general en el desenvolupament de programari d'oci, on els criteris d'accessibilitat com els recollits a les pautes WCAG per a contingut web [17], traslladables conceptualment a aplicacions

d'escriptori sovint s'incorporen tard en el cicle de desenvolupament, si és que s'hi arriben a incorporar. El present treball busca, en canvi, integrar l'accessibilitat com a requisit de disseny des de la concepció inicial del sistema.

Pel que fa a la interacció amb altres jugadors, plataformes com Chess.com incorporen elements socials llistes d'amics, rànquings i xats que enriqueixen l'experiència més enllà de la partida en si, mentre que aplicacions locals com Chess Titans o Lucas Chess centren la interfície exclusivament en el tauler i els controls d'un sol dispositiu, sense cap component social ni en xarxa. El projecte desenvolupat en aquest treball se situa en un punt intermedi: incorpora un mode de joc en xarxa entre dos ordinadors a diferència dels jocs purament locals, però sense els elements socials propis de plataformes web amb comptes d'usuari, atès que aquest aspecte queda fora de l'abast definit pels objectius inicials del projecte.

3 METODOLOGIA

3.1 Entorn de desenvolupament i tecnologies

El desenvolupament s'ha dut a terme sobre el sistema operatiu Microsoft Windows, utilitzant Microsoft Visual Studio [12], com a entorn integrat de programació, compilació i depuració. El llenguatge principal és C++, escollit per ser un llenguatge d'alt rendiment adequat per a aplicacions gràfiques en temps real, que permet un control detallat de la memòria i dels recursos del sistema i que és àmpliament utilitzat en el desenvolupament de motors gràfics i videojocs. L'experiència prèvia de l'autor amb aquest llenguatge ha permès centrar els esforços en la implementació funcional del sistema.

Per a la representació gràfica s'utilitza OpenGL, una API estàndard [1], [2], multiplataforma per al desenvolupament d'aplicacions 2D i 3D, escollida pel seu ús extensiu en entorns acadèmics i professionals i pel control detallat que ofereix sobre el pipeline gràfic. OpenGL permet implementar el tauler d'escacs en tres dimensions, la representació de les peces, la gestió de càmeres i la il·luminació de l'escena.

3.2 Planificació i gestió del projecte

El desenvolupament ha seguit una metodologia iterativa incremental [16], amb una combinació d'elements àgils i de cascada: el sistema s'ha construït progressivament mitjançant la implementació de funcionalitats en fases successives, validant cada component abans d'incorporar-ne de nous. El control de versions i la gestió de tasques s'han dut a terme mitjançant GitHub, que ha permès:

- Mantenir un historial complet de canvis sobre el codi font.
- Organitzar el desenvolupament en tasques (issues) associades a cada funcionalitat.
- Documentar el progrés de les diferents iteracions del projecte.
- Disposar d'un sistema de còpia de seguretat remot davant de possibles pèrdues de dades.

La taula II resumeix, de manera sintètica, les fites principals assolides al llarg del calendari de desenvolupament, des de la preparació de l'entorn gràfic fins a la implementació del joc en xarxa. El detall complet del calendari setmanal i de la taula de seguiment de proves es recull a l'apèndix.

TAULA II FITES PRINCIPALS PER PERÍODE DE DESENVOLUPAMENT

Període	Fita principal assolida
Març	Preparació de l'entorn gràfic, primeres càmeres i il·luminació, models de Blender
Abril–Maig	Lògica de moviments per classes, reconeixement de veu (Vosk) i moviments especials (enroc, promoció)
Maig–Juny	Sistema de torns i càmeres, il·luminació general, sistema d'ajuda per àudio (Piper)
Juny	Implementació del sistema de joc en xarxa amb servidor extern intermedi i sincronització TCP entre dos clients

En conjunt, la planificació inicial s'ha seguit amb fidelitat raonable: el retard detectat en una primera fase del projecte es va recuperar progressivament en les fases posteriors, i les funcionalitats de control per veu, assistència per àudio i joc en xarxa previstes per a les darreres fases del calendari s'han pogut completar dins del termini establert. Els objectius principals definits per al desenvolupament del sistema s'han assolit satisfactòriament i

el projecte s'ha mantingut alineat amb la planificació prevista per a la recta final del desenvolupament.

3.3 Reconeixement de veu

Per implementar el control per veu calia escollir una llibreria de reconeixement compatible amb l'entorn C++/OpenGL. Es van considerar tres opcions: Vosk, una llibreria de codi obert [3], [4], que funciona en mode fora de línia, compatible amb C/C++ i adequada per a vocabularis reduïts; whisper.cpp, una implementació en C++ del model Whisper, amb una precisió elevada però amb un cost computacional més alt i pensada per a dictat continu; i PocketSphinx, una llibreria més antiga desenvolupada per la Carnegie Mellon University, amb menor precisió que les alternatives actuals.

Finalment es va escollir Vosk, atès el bon equilibri que ofereix entre precisió, consum de recursos i facilitat d'integració amb C++. El fet de funcionar fora de línia i de permetre definir vocabularis reduïts la fa especialment adequada per al control per veu d'un joc d'escacs, en què les ordres possibles moviments de peces i coordenades del tauler són limitades i predictibles.

Aquesta decisió s'ha revelat encertada al llarg del desenvolupament: en funcionar íntegrament fora de línia, Vosk no introdueix dependències de connectivitat ni latències de xarxa en el reconeixement, un aspecte especialment rellevant en un sistema pensat per a usuaris que en depenen com a únic mecanisme de control. La possibilitat de definir un vocabulari reduït i específic per al domini dels escacs també ha facilitat, en fases posteriors, la implementació del sistema de normalització de comandes descrit a la secció 5.3.

4 DESENVOLUPAMENT

Aquesta secció descriu la implementació tècnica del sistema: l'organització del codi, la gestió dels models gràfics, el funcionament dels moviments i del control per veu, el sistema d'assistència per àudio, el joc en xarxa i la interfície d'usuari.

4.1 Arquitectura i organització del codi

El projecte s'estructura mitjançant una arquitectura modular, amb el codi dividit en fitxers independents cadascun en parella .h/.cpp. Entre els mòduls més rellevants destaquen:

- **Camera i GameObject**, encarregats de la gestió de la vista i de les transformacions (posició, rotació, escala) dels objectes de l'escena;
- **Il·luminació**, que defineix les fonts de llum, ombres i propietats dels materials per millorar la visibilitat i el realisme de l'escena;
- **ModelManager** i la jerarquia de classes de peces, que centralitzen la càrrega dels models 3D i encapsulen la lògica de moviment de cada peça;
- **Game i Board**, que coordinen l'estat global de la partida, els torns i la validació de regles;
- **AudioSystem**, que gestiona el sistema d'assistència per veu mitjançant una cua de missatges thread-safe;
- **NetworkSystem i Player**, responsables de la comunicació client-servidor mitjançant sockets TCP (Winsock2) i de la informació associada a cada jugador en el mode multijugador.

Aquesta organització afavoreix la reutilització del codi i facilita la incorporació de noves funcionalitats en fases posteriors del desenvolupament.

Per fer explícita l'estructura orientada a objectes del nucli del joc, la figura 1 resumeix la jerarquia d'herència emprada per a les peces. La classe base *Piece* defineix l'estat comú i la interfície de moviment, mentre que cada subclasse especialitza el càlcul de moviments legals segons les regles pròpies de cada peça.

Piece (classe base) x, y, color, moved getMoves(Board*)	
Pawn Peó	Rook Torre
Knight Cavall	Bishop Alfil
Queen Reina	King Rei

Fig. 1. Jerarquia d'herència de les classes de peces.

4.2 Representació gràfica i models 3D

La representació visual es basa en models 3D creats amb Blender [18], gestionats mitjançant la classe *ModelManager*. Cada element del joc tauler i peces es representa com una instància de *GameObject*, que conté la transformació de l'objecte (posició, rotació i escala) i una referència al model corresponent, cosa que permet reutilitzar models sense duplicar memòria i manipular cada peça de manera independent dins l'escena. El pipeline gràfic d'OpenGL s'utilitza per gestionar la rendització dels objectes, el sistema de càmera i la il·luminació bàsica, garantint una representació visual funcional i extensible (figura 2).

Aquesta separació entre la lògica del joc (*Game*, *Board*, classes de peces) i la seva representació gràfica (*GameObject*, *ModelManager*) permet que els canvis en l'estat lògic del tauler resultants d'una comanda de veu es tradueixin de manera transparent en actualitzacions de l'escena 3D, sense acoblar excessivament ambdues capes del sistema.

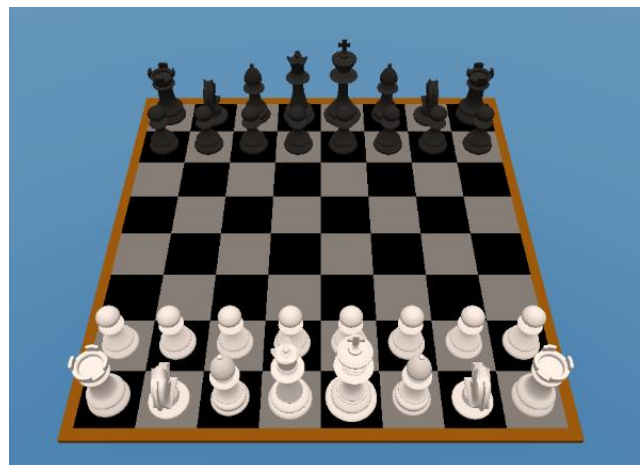


Fig. 2. Representació 3D del tauler i les peces a l'inici de la partida.

4.3 Sistema de moviments i control per veu

El flux de moviment segueix una arquitectura orientada a objectes en què cada peça encapsula el seu propi comportament, mentre que Game i Board actuen com a coordinadors. L'entrada del jugador es processa mitjançant Vosk, que genera una cadena de text normalitzada per adaptar errors habituals de pronunciació i convertir expressions orals en coordenades vàlides (per exemple, “a dos” o “E quatre”). La funció `extractSquare()` identifica la casella en format estàndard (A1–H8) i `parsePos()` la converteix a coordenades internes.

El sistema funciona en dos estats: selecció de peça i execució del moviment. Sense cap peça seleccionada, la comanda de veu s'interpreta com una selecció, comprovant que hi hagi una peça pròpia a la casella indicada; un cop seleccionada, la comanda següent es tracta com un intent de moviment, validant-lo mitjançant `getMoves(Board*)`, gestionant les captures i actualitzant tant l'objecte visual per interpolació com l'estat lògic del joc.

El sistema també incorpora els moviments especials. L'enroc (figura 3) requereix que es compleixin tres condicions: que el rei i la torre implicats no s'hagin mogut prèviament, que no hi hagi cap peça entre tots dos, i que el rei no estigui en escac ni passi per caselles atacades. Quan es detecta un enroc vàlid, es mou el rei dues caselles cap a la torre i aquesta es col·loca a la casella adjacent en direcció oposada, actualitzant-se de manera atòmica tant les posicions lògiques com els objectes visuals.

La promoció del peó es desencadena quan aquest arriba a l'última fila: el flux de moviment es bloqueja (`waitingPromotion`) fins que l'usuari indica per veu la peça desitjada (torre, cavall, alfil o reina), moment en què es reemplaça la instància de la peça i s'actualitzen el tauler lògic i l'objecte visual. En aquest cas, el canvi de torn es retarda fins que s'ha triat la nova peça; en la resta de moviments vàlids, es canvia el torn, s'actualitza la càmera segons el jugador actiu i es reinicia l'estat de selecció.

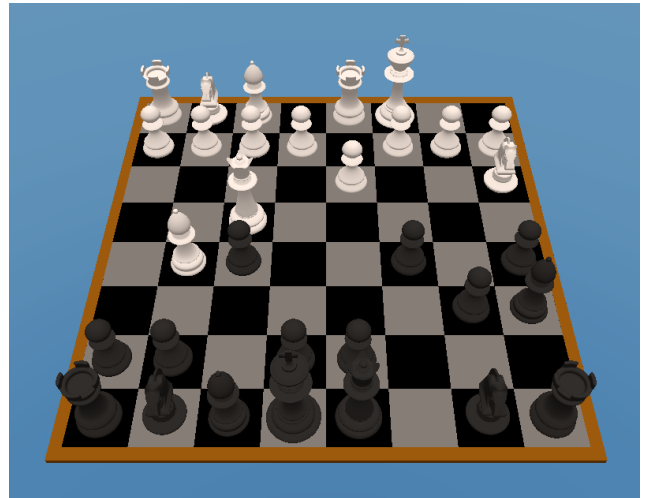


Fig. 3. Posició de partida amb l'enroc curt ja executat pel jugador de blanques.

4.4 Sistema d'assistència per àudio

Amb l'objectiu de millorar l'accessibilitat per a persones amb discapacitat visual, s'ha desenvolupat un sistema complet de retroalimentació per àudio mitjançant Piper [5], [6], un motor de síntesi de veu (TTS) de codi obert que funciona fora de línia i genera àudio en català amb el model de veu `ca_ES` (variant `upc_ona`). Internament, Piper s'executa des del seu propi directori d'instal·lació, rebent el text a sintetitzar per l'entrada estàndard i generant un fitxer d'àudio que es reproduïx posteriorment. La classe `AudioSystem` gestiona tota la comunicació amb Piper mitjançant una cua de missatges `thread-safe`, de manera que la generació i reproducció de veu no bloqueja ni el renderitzat gràfic ni el reconeixement de veu.

S'han definit funcions d'anunci específiques per a cada tipus d'esdeveniment de la partida:

- **Selecció de peça:** indica quina peça ha estat seleccionada i a quina casella es troba (p. ex. “Peó blanc a E2 seleccionat”).
- **Moviment i captura:** informa de la peça, la casella d'origen i la de destinació, i especifica quan s'ha capturat una peça enemiga.
- **Canvi de torn:** anuncia de quin color de peces és el torn actual.
- **Escac i escac i mat:** alerta l'usuari quan el rei és en escac i indica el guanyador en cas d'escac i mat.

- **Promoció:** indica que cal escollir per veu la nova peça (torre, cavall, alfil o reina).
- **Resum del tauler:** mitjançant la comanda “resum”, el sistema descriu totes les peces presents, el seu tipus, color i casella, oferint una “fotografia” auditiva completa de la partida en qualsevol moment.

Aquesta funcionalitat converteix la interacció amb el joc en una experiència plenament accessible mitjançant el so, complementant el control per veu i reforçant l'objectiu d'accessibilitat del projecte per a persones amb discapacitat visual.

4.5 Joc en xarxa entre dos ordinadors

El mode multijugador en xarxa s'implementa mitjançant la classe `NetworkSystem`, que encapsula la comunicació amb sockets TCP (`Winsock2`) [19]. La versió inicial feia que un dels ordinadors dels jugadors actués com a servidor i l'altre com a client; aquesta solució funcionava en una xarxa local, però depenia de la configuració del router i podia fallar quan calia travessar NAT o tallafocs.

- Per superar aquesta limitació, la versió final incorpora un servidor extern intermedi, implementat com un executable de consola independent (`ChessRelayServer`). En aquesta arquitectura cap jugador no obre connexions entrants: tots dos clients es connecten cap al servidor extern i aquest assigna automàticament els rols de color (WHITE per al primer client i BLACK per al segon).
- El servidor actua com a relay: rep missatges d'un client i els reenvia a l'altre mantenint el mateix protocol de longitud més missatge utilitzat pel joc. D'aquesta manera es transmeten moviments (MOVE:A2:A4), missatges d'àudio, sincronització d'inici de partida (READY), promocions, situacions d'escac o escac i mat i canvis de torn sense modificar la lògica principal del tauler.

Aquesta decisió redueix l'acoblament entre la capa de joc i la capa de comunicació. El client continua processant els missatges remots mitjançant cues thread-safe i aplicant-los sobre el tauler local, mentre que el servidor només coneix el protocol

de xarxa i no necessita interpretar les regles dels escacs.

El resultat és una arquitectura més robusta i justificable per a proves entre ordinadors diferents: els jugadors només necessiten conèixer la IP del servidor extern, i els routers dels jugadors no han de redirigir ports cap als seus equips. Aquesta modificació resol la principal limitació detectada en la primera versió del mode en xarxa.

4.6 Interfície d'usuari i sistema de càmeres

La interfície es basa en un entorn tridimensional amb OpenGL on es representen el tauler i les peces, complementat amb una graella de referència que identifica columnes i files. En el context d'aquest projecte, però, la interfície gràfica no es concep com el mecanisme principal d'interacció, sinó com un sistema de suport visual: la interacció directa es realitza mitjançant veu, mentre que la part gràfica actua com a element de retroalimentació que permet confirmar visualment les accions executades. Els elements principals que es representen són el tauler i les peces diferenciades clarament per model i color, el torn de joc i la quadrícula de suport amb el nom de cada columna i fila.

Un dels elements més rellevants és el sistema dinàmic de càmeres (figura 4): la càmera principal s'adapta automàticament en funció del torn de joc, situant-se en la posició òptima per al jugador actiu i millorant la visibilitat de les seves peces. Addicionalment, s'ha incorporat un mode de control manual mitjançant la tecla X que permet alternar entre vistes predefinides (zenital, perspectiva general o vistes laterals), útil especialment en situacions d'anàlisi o revisió de jugades.

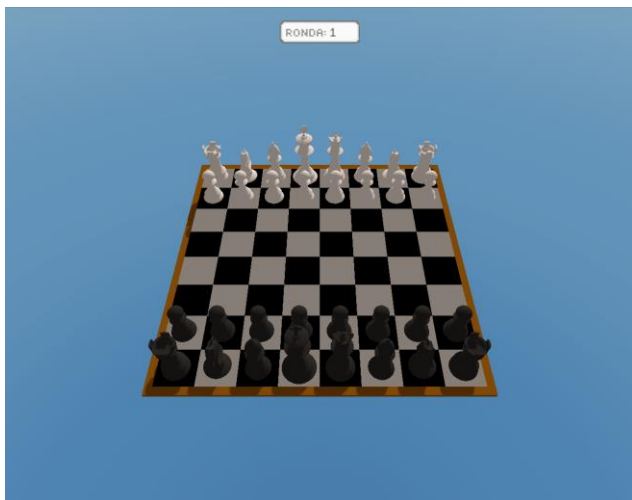


Fig. 4. Canvi automàtic de perspectiva de la càmera segons el torn de joc.

El veritable nucli del sistema d'interacció és, doncs, el control per veu, que permet a l'usuari jugar sense necessitat d'utilitzar extremitats superiors ni dispositius de precisió. Aquest enfocament persegueix quatre objectius principals: millorar l'accessibilitat per a persones amb discapacitat visual, reduir la dependència de perifèrics d'entrada tradicionals, explorar noves formes d'interacció natural en entorns 3D, i proporcionar informació auditiva contínua sobre l'estat de la partida.

Més enllà de la selecció i el moviment de peces, el disseny del sistema de veu deixa espai per ampliar el vocabulari de comandes en fases futures, incloent-hi possibilitats com:

- Informar per veu de les jugades realitzades per l'altre jugador en temps real, més enllà dels anuncis actuals.
- Gestionar la partida mitjançant ordres vocals addicionals (iniciar, aturar o reiniciar la partida).
- Facilitar la navegació entre diferents modes de joc local, en xarxa o, en el futur, contra la màquina sense necessitat de menús gràfics.

4.7 Motor d'intel·ligència artificial (treball futur)

La integració de Stockfish [10], [11], s'ha analitzat com a extensió per habilitar un mode de joc contra la màquina. Aquest motor es comunica amb les aplicacions mitjançant el protocol UCI, fet que permetria enviar l'estat del tauler i rebre el moviment

calculat pel motor. En aquest TFG no s'ha implementat perquè era un objectiu de baixa prioritat, però l'arquitectura actual permetria incorporar-lo en futures iteracions com un procés extern, de manera similar a la integració realitzada amb Piper.

5 RESULTATS

5.1 Grau d'assoliment dels objectius

Els objectius d'alta prioritat s'han assolit íntegrament: el sistema permet disputar una partida completa d'escacs en un entorn 3D, amb tots els moviments de les peces inclosos l'enroc i la promoció i una gestió de torns integrada amb el sistema de càmeres. També s'han assolit els objectius de prioritat mitjana: el control per veu amb Vosk permet jugar una partida completa únicament mitjançant ordres parlades, el sistema d'assistència per àudio amb Piper ofereix retroalimentació contínua, i el joc en xarxa client-servidor permet disputar partides sincronitzades entre dos taulers. L'únic objectiu no assolit correspon a la prioritat baixa: la integració d'un motor d'intel·ligència artificial (Stockfish) per jugar contra la màquina, que es planteja com a treball futur.

Aquest grau d'assoliment és coherent amb l'ordre de prioritats establert a la secció 1.1: les funcionalitats classificades com a nucli funcional i com a prioritat mitjana precisament aquelles que defineixen l'accessibilitat del sistema s'han completat dins del termini, mentre que l'única mancança correspon a una funcionalitat explícitament classificada com a no essencial des de l'inici de la planificació que es mostra a la taula III.

TAULA III GRAU D'ASSOLIMENT DELS OBJECTIUS

Objectiu	Prioritat	Estat
Lògica completa i representació 3D	Alta	Assolit
Interfície bàsica i partides locals	Alta	Assolit
Control per veu (Vosk)	Mitjana	Assolit
Joc en xarxa amb servidor extern	Mitjana	Assolit
Assistència per àudio (Piper)	Mitjana	Assolit
Motor d'IA (Stockfish)	Baixa	No assolit
Poliment estètic i de rendiment	Baixa	Parcial

5.2 Validació i proves

La validació del sistema s'ha realitzat mitjançant proves manuals centrades en la càrrega dels models 3D, el comportament de la càmera, l'execució dels moviments per veu incloent enroc i promoció i la sincronització de moviments, àudio, promoció, escac/escac i mat i canvis de torn entre dos clients connectats al servidor extern. El detall de les proves es recull a l'apèndix, incloent-hi la validació específica del servidor extern desenvolupat.

5.3 Limitacions i problemes coneguts

Durant el desenvolupament s'han resolt problemes relacionats amb el reconeixement de veu, la representació visual, la portabilitat i la comunicació en xarxa. Les confusions fonètiques de Vosk (per exemple, la lletra C interpretada com "fer" o "ser") s'han mitigat amb un sistema de normalització de comandes; els problemes de materials i shaders dels models s'han corregit ajustant la gestió visual de les peces; i l'execució fora de Visual Studio s'ha resolt mitjançant rutes relatives i una estructura portable de carpetes. Un problema a mencionar és a l'hora d'utilitzar la llengua catalana que no està tan avançada en ajudes d'àudio o reconeixement per veu com l'anglès o el castellà.

- La limitació més rellevant del mode en xarxa era que la primera arquitectura, on un jugador actuava com a servidor, depenia de routers, NAT i tallafocs. La versió final incorpora un servidor extern intermedi que rep connexions sortints dels dos clients i reenvia els missatges de partida. Com a treball futur, aquest servidor es podria desplegar de forma permanent en un VPS i ampliar-se amb sales múltiples, reconexió automàtica, autenticació i mecanismes bàsics de seguretat.

6 CONCLUSIONS I TREBALL FUTUR

Aquest treball ha demostrat la viabilitat de desenvolupar un joc d'escacs 3D amb OpenGL on la veu és el canal principal de control. El sistema permet disputar partides completes amb lògica d'escacs, assistència auditiva i sincronització entre dos ordinadors mitjançant un servidor extern intermedi,

responent als objectius d'accessibilitat plantejats inicialment.

Com a treball futur es planteja integrar Stockfish mitjançant UCI, desplegar el servidor en un VPS o servidor institucional i ampliar la xarxa amb sales múltiples, reconexió automàtica i mesures bàsiques de seguretat. L'enfocament desenvolupat també es podria extrapolar a altres jocs de tauler o aplicacions interactives on l'accessibilitat encara és limitada.

ÚS DE LA IA GENERATIVA

S'han emprat eines d'intel·ligència artificial generativa com a suport puntual per a tasques de redacció, síntesi i formatació, sempre sota revisió de l'autor. El contingut tècnic, el disseny, la implementació i els resultats corresponen al treball desenvolupat durant el TFG.

AGRAÏMENTS

L'autor vol agrair al tutor, Javier Sanchez Pujadas, l'orientació i el seguiment durant el projecte, així com el suport rebut per part de família i amistats al llarg del procés.

BIBLIOGRAFIA

- [1] Khronos Group, “OpenGL Overview.” [En línia]. Disponible: <https://www.opengl.org/>
- [2] Khronos Group, “OpenGL Registry and Specifications.” [En línia]. Disponible: <https://registry.khronos.org/OpenGL/>
- [3] Alpha Cephei, “Vosk Offline Speech Recognition API.” [En línia]. Disponible: <https://alphacephei.com/vosk/>
- [4] Alpha Cephei, “Vosk API — GitHub Releases.” [En línia]. Disponible: <https://github.com/alphacep/vosk-api/releases>
- [5] Rhasspy, “Piper — A fast, local neural text to speech system.” [En línia]. Disponible: <https://github.com/rhasspy/piper>
- [6] Rhasspy, “Piper Voices.” [En línia]. Disponible: <https://huggingface.co/rhasspy/piper-voices>
- [7] espeak-ng. [En línia]. Disponible: <https://github.com/espeak-ng/espeak-ng>
- [8] PortAudio, “Audio I/O Library.” [En línia]. Disponible: <https://files.portaudio.com/>
- [9] Chess Programming Wiki. [En línia]. Disponible: <https://www.chessprogramming.org/>
- [10] Stockfish, “Strong open source chess engine.” [En línia]. Disponible: <https://stockfishchess.org/>
- [11] Chess Programming Wiki, “Stockfish.” [En línia]. Disponible: <https://www.chessprogramming.org/Stockfish>
- [12] Microsoft, “Visual Studio Community.” [En línia]. Disponible: <https://visualstudio.microsoft.com/vs/community/>
- [13] Chess.com. [En línia]. Disponible: <https://www.chess.com/>
- [14] Lucas Chess. [En línia]. Disponible: <https://lucaschess.pytho-nanywhere.com/>
- [15] Free3D, “Models 3D d’escacs.” [En línia]. Disponible: <https://free3d.com/es/modelos-3d/ajedrez>
- [16] Asana, “Project management methodologies.” [En línia]. Disponible: <https://asana.com/es/resources/project-management-methodologies>
- [17] W3C, “Web Content Accessibility Guidelines (WCAG) 2.1.” [En línia]. Disponible: <https://www.w3.org/TR/WCAG21/>
- [18] Blender Foundation, “Blender.” [En línia]. Disponible: <https://www.blender.org/>
- [19] Microsoft, “Winsock2 — Windows Sockets 2.” [En línia]. Disponible: <https://learn.microsoft.com/windows/win32/winsock/>
- [20] Curs d’introducció a OpenGL en C++, YouTube. [En línia]. Disponible: <https://www.youtube.com/watch?v=W3gAzLwflPO>

APÈNDIX

A1. PLANIFICACIÓ TEMPORAL DETALLADA

La taula IV detalla, per mes, les principals tasques desenvolupades al llarg del projecte, complementant el resum de fites presentat a la secció 3.2 i incorporant la fase final de servidor extern desenvolupada entre el 26 i el 28 de juny.

TAULA IV CALENDARI DETALLAT DE DESENVOLUPAMENT

Mes	Tasques desenvolupades
Març	Preparació de la part gràfica (càmeres provisionals, primera il·luminació, UI bàsica per a proves), cerca i càrrega de models de Blender i col·locació de les peces al tauler.
Abril	Implementació dels moviments del peó (avenç d'una o dues caselles, captura en diagonal) amb moviment interpolat i fluid.
Maig	Organització del codi per classes, integració de Vosk i PortAudio, implementació dels moviments de totes les peces, l'enroc, la selecció visual, el sistema de torns i la col·locació de les peces negres.
Juny	Millora del reconeixement de comandes de lletres i números, sistema d'il·luminació general, sistema d'ajuda per àudio (Piper), primera implementació client-servidor i, del 26 al 28 de juny, desenvolupament del servidor extern intermedi per superar les limitacions dels routers i sincronitzar dos clients.

Data	Prova	Resultat esperat
10-28 juny	Creació del servidor extern intermedi (ChessRelayServer)	El servidor accepta dos clients, assigna rols WHITE/BLACK i manté oberta la sessió de partida
10-28 juny	Relay TCP de missatges de partida	Els missatges MOVE, AUDIO, READY, promoció, escac i escac i mat es reenvien correctament entre els dos jugadors
10-28 juny	Prova del servidor amb dos ordinadors	Els dos clients es connecten a la IP del servidor i la partida queda sincronitzada sense connexions entrants als ordinadors dels jugadors

A2. SEGUIMENT DE PROVES

La taula V recull una selecció representativa de les proves de validació dutes a terme durant el desenvolupament, incloent-hi la data, l'objecte de la prova i el resultat esperat. Les darreres files documenten les proves específiques del servidor extern.

TAULA V SELECCIÓ DE PROVES DE VALIDACIÓ REALITZADES

Data	Prova	Resultat esperat
14–15 mar	Preparació de la part gràfica	Interfície gràfica acabada per poder començar a treballar
21 mar	Càmeres provisionals i il·luminació	Observar i provar els models amb profunditat visual
28–29 mar	Càrrega i col·locació de models	Tauler horitzontal i peces correctament posicionades
5 abr	Moviment del peó (1/2 caselles i captura)	Moviment inicial i captura en diagonal correctes
1 maig	Estructuració del codi i classe Board	Codi organitzat per classes; tauler i peces gestionats correctament
1 maig	Vosk i vcpkg PortAudio	Reconeixement de veu i àudio funcionant correctament
2–3 maig	Selecció i moviment de peces per veu	Selecció i moviment correctes per a totes les peces
9 maig	Enroc i millora de comandes lletra/número	Moviment d'enroc correcte i reconeixement més robust
19–21 maig	Sistema de torns, càmera i il·luminació	Canvi de torn i de perspectiva correcte; escena més realista
30 maig – 6 jun	Sistema d'ajuda per àudio	Moviments i accions anunciats correctament per veu
6–9 jun	Sistema client-servidor (mateix PC)	Connexió i sincronització de moviments entre dues instàncies