

Projecte De Programació

Chess game 1^a entrega

Daniel Palomo Cabrera	daniel.palomo
Arnau Santos Fernandez	arnau.santos.fernandez
David Soldevila Puigbi	david.soldevila.puigbi

Versió 1.0

Índex:

Especificació de classes.....	3
Llista de repartició de treball.....	6
Algorismes.....	7
Especificació de casos d'ús.....	8
Diagrama de classes	9
Diagrama de casos d'us.....	10

Especificació de classes

Problem

Classe per representar un problema. Permet la visualització i modificació de les dades, la verificació del mateix i la gestió del ranking de jugadors d'aquell problema.

GameAbs

Classe per gestionar un joc. Permet veure les peces que té la taula i moure-les.

Game

Classe per gestionar un joc on hi juga mínim una persona. Aquesta té en compte el temps de decisió per fer les rankings dels problemes.

AICompetition

Classe per gestionar perides entre les IAs. Es juguen K partides i després es poden adquirir els resultats, nombre de partides guanyades per cada color.

Player

Classe per representar els jugadors (usuaris). Guarda d'informació necessària per representar un usuari i permet calcular el seu ELO.

Human

Classe per representar un jugador humà. Aquest permet guardar una password per autenticar-se.

AI

Classe que conte l'algoritme del jugador maquina.

AI1

Classe amb l'algorisme de IA de dificultat fàcil.

Board

Classe per representar un taulell d'escacs i gestionar-lo. Permet moure i afegir peces i codificar el contingut de la taula en codificació fen.

Piece

Classe per representar una peça. Dona informació dels possibles moviments que pot fer.

Té una subclasse per cada tipus de peça i una per representar una peça buida.

Totes les subclasses utilitzen el seu propi moviment per calcular els possibles posicions.

King

Classe que representa la peça Rei.

Pawn

Classe que representa la peça Peó.

Queen

Classe que representa la peça Reina.

Rock

Classe que representa la peça Torre.

Bishop

Classe que representa la peça Alfil.

nullPiece

Classe que representa una peça buida.

Knight

Classe que representa la peça Cavall.

CtrlDomainCreator

Controladora que s'encarrega de crear instancies.

CtrlDomainGame

Controladora que maneja el Game.

chessException

Classe derivada d'Exception la que utilitzem per llançar les nostres pròpies excepcions personalitzades.

Repartició de treball:

Daniel Palomo: Piece, King, Queen, Pawn , Rock, NullPiece, Bishop, Knight, Human, Player, ChessException, Board

David Soldevila: Problem, GameAbs ,Game ,CtrlDominiCreator

Arnau Santos: AI ,AI1 ,AICompetition_ ,CtrlDomainGame.

Algorismes:

Problem verifier

Per verificar un problema fem servir un algorisme DFS (Depth First Search) per buscar una possible taula en la qual al cap de N+1 torns s'arriba a matar el rei significa que existeix, com a mínim, una solució.

AI players

Les dues IAs implementen l'algorisme MinMax, que consisteix en la cerca en arbre consultant tots els possibles moviments de les peces, guiada per una funció heurística que retorna un valor en funció de la situació del taulell: les peces atacants sumen punts mentre que les peces defensores resten punts.

Player ranking

$$\text{Performance rating} = \frac{\text{Total of opponents' ratings} + 400(\text{Wins-Losses})}{\text{Number of Games}}$$

Problem Ranking

Es guarda una llista ordenada amb la llista amb la suma dels temps de cada jugada que ha fet i el nom del jugador.

FEN "encoder"

Es recorrerà la taula i es completa una String amb els espais en blanc i les peces.

FEN "decoder"

Es recorre el fen i es va completant la taula a la vegada es verifica que no s'accedeix el nombre màxim de fitxes d'un mateix tipus i que hi hagi exactament vuit cel·les representades en cada segment.

Especificació del cas d'us

- Profile management: Usuari pot decidir entre fer login o crear un nou usuari.
- Login: Usuari té que entrar les seves credencials o entrar com invitat.
- Signup: Usuari crea un nou perfil introduït un id i la contrasenya.
- Problem management: L'usuari pot carregar un problema o crear un nou.
- Create: Usuari podrà crear un problema des de zero.
- Modify: Usuari pot dissenyar un problema a partir d'un existent.
- Delete: Usuari pot esborrar un problema existent.
- Save: Usuari pot guardar un problema, independentment de si esta acabat o no.
- Verify: El usuari pot verificar que el problema es correcte i té solució.
- Free movement of pieces: Usuari pot moure peces allà on vulgui.
- Game management: Usuari pot posar les característiques del Game i jugar.
- Define players: Usuari pot definir les característiques dels players
- Define color: Usuari pot decidir el color de cada Player.
- Define player: el usuari pot definir quin tipus de Player jugarà .
- Player login 2: Un segon usuari pot iniciar sessió per jugar.
- Select IA: Usuari definirà la dificultat de la IA
- Load problem: Usuari pot seleccionar un problema que jugar des de la base de dades.
- Play problem: Usuari juga el Game.
- Restart: Usuari pot reiniciar el Game.
- Quit: Usuari pot abandonar el joc.
- Consultation: Usuari pot seleccionar entre estadístiques
- Problem statistics: usuari pot buscar estadístiques de problemes
- Player statistics: Usuari pot consultar estadístiques de Player.

