Implementarea unui joc de șah în C++

Acest document prezintă pașii pentru implementarea unui joc de șah în C++, inclusiv funcțiile și clasele necesare. Vom crea o clasă principală care reprezintă tabla de șah, piesele și regulile de bază ale mutării.

# 1. Clasa ChessBoard

Clasa ChessBoard va reprezenta tabla de șah și va conține funcții pentru a inițializa tabla, a afișa tabla și a verifica starea jocului.

Funcții propuse:

- void initBoard(): Inițializează tabla de șah cu piesele în poziția inițială.

- void displayBoard(): Afișează tabla de șah în consolă.

- bool isCheckmate(): Verifică dacă unul dintre jucători este în șah mat.

- bool isStalemate(): Verifică dacă jocul a ajuns la pat.

# 2. Clasa ChessPiece

Clasa ChessPiece va fi clasa de bază pentru toate piesele de șah (Rege, Regină, Nebun, Cal, Turn, Pion).

Funcții propuse:

- virtual bool isValidMove(int startX, int startY, int endX, int endY): Funcție virtuală care verifică dacă mutarea este validă pentru piesă.

- char getSymbol(): Returnează simbolul piesei pentru afișarea pe tablă.

# 3. Clasa Pawn (Pion)

Clasa Pawn va extinde clasa ChessPiece și va implementa mutările specifice pionului.

Funcții propuse:

- bool isValidMove(int startX, int startY, int endX, int endY): Verifică dacă mutarea este validă pentru un pion (înainte cu o poziție sau două la prima mutare, captură pe diagonală).

# 4. Clasa Rook (Turn)

Clasa Rook va extinde clasa ChessPiece și va implementa mutările specifice turnului.

Funcții propuse:

- bool isValidMove(int startX, int startY, int endX, int endY): Verifică dacă mutarea este validă pentru un turn (mișcare pe linii drepte).

# 5. Clasa Knight (Cal)

Clasa Knight va extinde clasa ChessPiece și va implementa mutările specifice calului.

Funcții propuse:

- bool isValidMove(int startX, int startY, int endX, int endY): Verifică dacă mutarea este validă pentru un cal (mișcare în formă de "L").

# 6. Clasa Bishop (Nebun)

Clasa Bishop va extinde clasa ChessPiece și va implementa mutările specifice nebunului.

Funcții propuse:

- bool isValidMove(int startX, int startY, int endX, int endY): Verifică dacă mutarea este validă pentru un nebun (mișcare pe diagonală).

# 7. Clasa Queen (Regină)

Clasa Queen va extinde clasa ChessPiece și va implementa mutările specifice reginei.

Funcții propuse:

- bool isValidMove(int startX, int startY, int endX, int endY): Verifică dacă mutarea este validă pentru o regină (mișcare pe linii drepte și diagonale).

# 8. Clasa King (Rege)

Clasa King va extinde clasa ChessPiece și va implementa mutările specifice regelui.

Funcții propuse:

- bool isValidMove(int startX, int startY, int endX, int endY): Verifică dacă mutarea este validă pentru un rege (mișcare cu o singură poziție în orice direcție).