# Sakk

## Fájlok:

- algebraic.c: Algebrai notációból vált mindkét irányba
- analysis.c: Visszajátszással kapcsolatos függvényeket tartalmaz
- board.c: Az aktuális játékkal kapcsolatos információ kezelő és segéd függvényeket tartalmaz
- buttons.c: Gombokkal kapcsolatos függvényeket tartalmaz
- check.c: Sakk és sakkmatttal kapcsolatos függvények
- chess.c: Fő fájl
- econio.c: Parancssorra írás
- graphics.c: Grafikus megjelenítéssel kapcsolatos függvények
- mouse.c: Egérrel kapcsolatos függvények
- moves.c: Lépésekkel kapcsolatos függvények
- piece.c: Bábukkal kapcsolatos függvények
- promotion.c: Átalakulással kapcsolatos függvények
- square.c: Egy adott négyzettel dolgozó függvények
- views.c: Menükkel kapcsolatos függvények

## Működő funkciók:

Teljes játék lejátszása
Játékállás mentése
Analízis, tetszőleges folytatás nélkül
Analízisnél a jobbra- és balra nyilakkal lehet előre és vissza lépni

## Fő adatstruktúrák

A jelenlegi játékkal kapcsolatos minden információt a Board struktúra tartalmaz. Ez többek között tartalmazza a bábuk helyzetét egy kétdimenziós tömbben, és az eddigi lépéseket. A lépéseket mindig csak sorban kell elérni, de mindkét irányban, így azt egy kétszeresen láncolt listában tárolja a program.
A sakk(matt)-ról a CheckData típus tárolja: sakkban van-e, hány bábu tartja sakkban, ez kettő lehet, és hogy ezek hol vannak. Mivel csak kettő lehet, dinamikus memóriakezelés nem szükséges.

# A program fontosabb függvényei

Board* newGameFromFen(const char *fenStr, SDL_Renderer *renderer)

Létrehoz egy új táblát, vagyis új játékot kezd, a FEN stringként megadott játék állással. Az új játék a megadott rendererre lesz rajzolva.

CheckDataByColor willNextMoveBeCheck(Board *board, Move move)

Meghatározza, hogy a megadott lépés sakkot fog-e eredményezni, és megadja hogy mely bábuk okozzák a sakkot.

bool isValidMove(Board *board, Move move, Square *enPassante, Move *rookMove, int *newCastlingAvailability)

Igaz értékkel tér vissza, ha a megadott lépés érvényes. enPassante által mutatott helyre írja azt a négyzetet, amelyre en passante lépésnél lépni lehet. Sáncolás esetén a rookMove által mutatott helyre írja a bástya lépését, newCastlingAvailabilty helyére pedig a módosított bitmezőt, ami leírja hogy melyik bástyákkal lehet sáncolni. A pointerek közül bármennyi lehet NULL.

void movePieceWithCheck(Board *board, Move move)

Végrehajtja a megadott lépést, ha az érvényes. Kezeli az átváltozást is. Frissíti a tábla belső állapotát.