

Connect4Game Tervezési minta

1. MVC szoftverarchitektúra:

A program MVC (Model-View-Controller) mintát követ, amit az alábbi módon lehet felosztani:

Model: (Adat és logika) Ez a rész kezeli az adatok tárolását és a program működésének szabályait.

View: (Nézet) Feladata az adatok megjelenítése és felelős a felhasználóval való interakciókért.

Controller: (Vezérlő) A Model és a View között közvetíti és feldolgozza a felhasználói bemeneteket. Tehát a feladata játékmenet irányítása, a játék logikájának és a felhasználói interakcióknak az összekapcsolása.

2. Az MVC tervezési minta a Connect4Game programra kivetítve

Model (Adatok és játékmenet logika)

- **Board osztály**

Szerepe: A játék tábla állapotát tárolja, és a játékmenethez kapcsolódó logikát kezeli (például bábú helyezése, győzelem ellenőrzése). Az MVC-ben ez a fő Model komponens, mert a tábla állapotát reprezentálja és a játékmenet szabályait tartalmazza.

A Board osztály feladatai: Tábla létrehozása vagy betöltése, bábú leletésének logikája, győzelmi feltételek ellenőrzése (vízszintesen, függőlegesen, átlósan), játék mentése.

- **DatabaseManager osztály**

Szerepe: Az adatbázissal való kommunikációért felel, például a játékosok győzelmeinek mentése és statisztikák lekérdezése. Az MVC-ben a Model része, mert az adatokat kezeli, tárolja és elérhetővé teszi a rendszer más részei számára.

A DatabaseManager osztály feladatai: Az adatbázis inicializálása és tábla létrehozása, győzelmi statisztikák frissítése, statisztikák lekérdezése és megjelenítése.

View (Megjelenítés és felhasználói interakció)

- **Player osztály**

Szerepe: A felhasználói interakciókat kezeli. Például a játékos nevét tárolja, bemeneteket kér a felhasználótól például oszlopválasztáskor. Az MVC-ben a View része, mert a játékos közvetlen interakcióit kezeli (bemenetek és hibakezelés).

A Player osztály feladatai: A játékos nevének megadása, a kiválasztott oszlop lekérése, hibás bemenetek kezelése és a játékos újabb választásra kérése.

- **Board osztály printBoard() metódusa**

Szerepe: A játék állapotának megjelenítése a képernyőn. Az MVC-ben ez a View egy része, mert vizuálisan reprezentálja a játék állapotát a felhasználó számára.

Controller (Irányítás és logika koordinálása)
--

- **GameController osztály**

Szerepe: A játék teljes logikájának és irányításának koordinálásáért felelős. Kezeli a játékosok lépéseit, a játék állapotát és a mentési/betöltési funkciókat. Az MVC-ben ez a Controller, mert közvetít a Model és a View között. A bemeneteket fogadja a Player osztálytól, és a Board logikáját használja a játék állapotának frissítésére.

A GameController feladatai: A játékos lépéseinek kezelése (saját lépés/képernyő frissítése), bábu leejtésének engedélyezése és győzelem ellenőrzése, játékosváltás és játék mentése.

2. Osztályok és kapcsolataik:

1. Player (Játékos)

Szerepe: A játékosok adatait tárolja, mint például a nevük és a használt szimbólumuk (pl. Y a sárga játékosnak, R a piros számítógépnek).

Kapcsolat: A Player osztály közvetlenül kapcsolódik a GameController osztályhoz, amely az aktuális játékost és annak választásait kezeli.

2. Board (Játéktábla)

Szerepe: A játéktábla tárolása és az azt manipuláló logika (például a leejtett bábuk kezelése és a győzelem ellenőrzése).

Kapcsolat: A Board osztály szorosan együttműködik a GameController-rel, aki a játékmenetet irányítja. A Board felelős a táblázat állapotának nyomon követéséért, és az interakciók kezeléséért (például leejtés, győzelem ellenőrzése).

3. GameController (Játékirányító)

Szerepe: A játékmenet logikáját irányítja. Kezeli a játékosok váltakozó lépéseit, a játék kezdését és végét, a mentést és betöltést, valamint a statisztikák frissítését.

Kapcsolat: A GameController összekapcsolja a Player, Board, és a DatabaseManager osztályokat. Ő felelős az interakciók koordinálásáért és a játékmenet irányításáért.

4. DatabaseManager (Adatbázis kezelő)

Szerepe: Az adatbázis műveletek végrehajtása, mint például a játékosok statisztikáinak tárolása és lekérdezése.

Kapcsolat: A DatabaseManager a GameController által használt osztály, és az adatok mentéséért és lekérdezéséért felelős.

3. Funkcionális minta:

Játék indítása és irányítása:

A GameController osztály indítja el a játékot a startGame metódussal.

A Board osztály inicializálja a játéktáblát.

A felhasználó nevét bekéri a GameController, és egy új Player objektumot hoz létre a játékos számára.

A program elindítja a játékot, és a játékosok váltakozva lépnek a táblán. Az első játékos a sárga (Y), a második (számítógép) a piros (R).

A játék folyamán a GameController osztály meghívja a Board osztály metódusait a játék állapotának frissítésére (pl. dropPiece, checkWin).

A DatabaseManager segít a játékosok győzelmeinek tárolásában, és statisztikák megjelenítésében.

Játék mentése és betöltése:

A játék során a GameController osztály lehetőséget biztosít arra, hogy a játékot egy fájlba mentse a Board osztály saveToFile metódusának segítségével.

A játék betöltésekor a Board osztály képes betölteni a korábbi mentést a loadFromFile metódusával.

4. Adatbázis kezelés:

A DatabaseManager felelős a játékosok győzelmeinek mentéséért és statisztikai adataik nyomon követéséért az adatbázisban. Az adatbázis kapcsolódik az SQLite-hoz, és tartalmaz egy players táblát, amely a játékosok nevét és győzelmeiket tárolja.

5. Tesztelés:

A programhoz különböző egységtesztek is tartoznak, hogy biztosítsák a helyes működést.

BoardTest: Ellenőrzi a táblával kapcsolatos műveletek helyes működését, például a bábu leesését és a teljes oszlopok kezelését.

DatabaseManagerTest: Teszteli az adatbázis műveletek, mint például a győzelmek hozzáadását és az adatbázis állapotának lekérdezését.

PlayerTest: Ellenőrzi a játékosok helyes inicializálását és a szimbólumok kezelését.

6. Következő lépések az MVC minta alapján:

A program bővítéséhez és fejlesztéséhez az alábbi lehetőségek merülhetnek fel:

- AI fejlesztése: A számítógép jelenlegi döntései véletlenszerűek, egy fejlettebb mesterséges intelligencia implementálása javíthatja a játékelményt.
- Felhasználói felület: A játéknak lehetne egy grafikus felhasználói felülete (GUI), amely interaktívabbá tenné a játékot.
- Hálózati játék: Lehetőség adható a többjátékos online játéknak.

7. Következtetés:

A program különböző komponensekre van bontva, amelyek meghatározzák a játékmenetet, az adatkezelést és a felhasználói interakciókat. Ez a struktúra segít a kód olvashatóságának és karbantarthatóságának javításában, valamint megkönnyíti a jövőbeli bővítéseket.