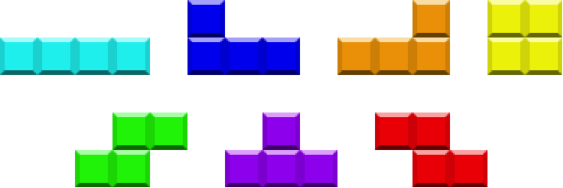


2 DIMENZIÓS TÖMBÖK ALKALMAZÁSA

# Leírás

A **Tetris**([oroszul](https://hu.wikipedia.org/wiki/Orosz_nyelv): Тетрис) egy ügyességi és logikai videójáték, melynek első változatát az orosz [Alekszej Leonyidovics Pazsitnov](https://hu.wikipedia.org/wiki/Alekszej_Leonyidovics_Pazsitnov) tervezte meg és készítette el. A játék 7 féle külömbözö elemet tartalmaz, melyek egy 20 x 10 – es játéktáblán helyezkednek el. Az elemek négyzetről-négyzetre mozognak lefele, és e közben a játékosnak leheősége van jobbra illetve balra mozgatni a színes elemek, illetve elforgathatja őket. Ha a játékosnak sikerül egy folytonos sort létrehoznua, azon sor törlődik és a hozzá rendelt érték hozzáadódik a játékos pontszámához. A játék célja, hogy a játékos minél több sort kitöltsön, ezzel pontokat szerezve, mielőtt betelik a játéktábla.

A 7 elem mindegyike 4 kis négyzetből tevődik össze, van köztük teljesen szimmetrikus forma is (négyzet), olyan alaakzat, amlyet ha 2x forgatunk el 90 fokkal akkor visszakapjuk az eredeti alakzatot és olyan, melyet csak 4 forgatással tudunk visszaállítani eredeti állapotába. A színük a játék verziójától függ, tetszőleges, viszont az újabb verziókba, a színes ábrázolás lehetőségének köszönhetően, mindegyik elem külömböző színű.



Egy lehetséges ábrázolása a 7 elemnek

# Feladat

Implementáljuk a Tetris játékot C/C++ nyelvben.

# Elképzelés

Felhasználva a graphics.h grafikus könyvtárat, létrehozni a játékot C++ nyelvben úgy, hogy a randomizálts sorrendben következő alakzatok végleges pozicióját jelöljük a mátrixban. Ha az objektum koordniátáit megfeleltetjük a matrix sor- és oszlop-indexének, valamint ha a megfelelő poziciókban az alakzat színkódját tároljuk el, kirajzolható a Tetris játék pillanyatni helyzete. A graphics.h könyvtár segítségével megoldható a grafikus ablak létrehozása, ezen ablak törlése és újrarajzolása, az úgynevezett “flickering” kiküszöbölése, az alakzatok billentyűleütésre való forgatása és mozgatása. Ezen kívül egy koordinatak struktúra (x és y mezővel rendelkező struktúra) felhasználásával könnyen tudjuk követni egy alakzat állapotát a kezdeti, azaz a generálási ponttól, a végsőig, vagyis ahhonnan már nem tud tovább mozogni és “átmásolódik” a mátrixba. Ezen kívül felhasználunk többféle típusú segédváltozót, mint például az eltolás jobbra vagy balra, hányszor volt fordulás kérve, egy karakterláncot a pontszám és egyéb karaktersor kiírásához.

# 2 dimenziós tömb (Mátrix) adatszerkezet és a tárolt adatok

A feladat megoldásához és természetes implemetálásához a legkézenfekvőbb adatszerkezet a mátrix, azaz a 2 dimenziós tömb lenne.

# Mátrixról általánosan

A mátrix egy 2 dimenziós tömb, amely azonos típusú elemek sorszámozott tárolója. A mátrixhoz tartozó két fontos információ a sorok száma (általában “n” -el jelöljük) és az oszlopok száma (általában “m” -el jelöljük), viszont abban az esetben ha sorfolytonos ábrázolást választunk, akkor elegendő megadni csak egy értéket, a tömb hosszát, amely tulajdonképpen az előbbi kettő szorzata. Az elemeket ebben az esetben két index vagy egy index segítségével érjhetük el, ábrázolástól függően. Az elemek lehetnek egyszerűek vagy akár összetettek is.

A tömb < index,elem > párok halmaza, ahol minden létező indexnek megfelel egy elem.

# Mátrixok alkalmazása

A mátrixokat széles körben alkalmazzák, mivel akár táblázatszerűen tárolhatóak el benne az információk. Mátrixokat szoktak használni [lineáris egyenletek](https://hu.wikipedia.org/wiki/Line%C3%A1ris_egyenlet) és [lineáris](https://hu.wikipedia.org/wiki/Line%C3%A1ris_lek%C3%A9pez%C3%A9s), valamint [bilineáris függvények](https://hu.wikipedia.org/w/index.php?title=Biline%C3%A1ris_f%C3%BCggv%C3%A9ny&action=edit&redlink=1) leírására. A mátrixok – a [lineáris algebra](https://hu.wikipedia.org/wiki/Line%C3%A1ris_algebra) (egyik) leghasznosabb fogalmaként – a matematikának a gyakorlatban legtöbbször alkalmazott eszközei között vannak, a matematika számos más ága mellett pedig a fizikától és komputergrafikától kezdve a biológián át egészen a nyelvészetig, számtalan tudományágban használhatóak akár az elméleti leírás tömör megfogalmazására, akár a számítások megkönnyítésére vagy automatizálására. Mivel a pixelek ábrázolására is praktikus ez az adatszerkezet, nagyon sok számítógépes játékhoz alkalmazzák, képek ábrázolásához, forgatásához, tükrözéséhez, ezen kívül permutációk és transzformációk adatainak eltárolásához is felhasználják.

# Mátrix adatszerkezet – eltárolt információk

Az adatszerkezetben el kell tárolnunk egész típusú értékeket, melyek azt fogják szimbolizálni, hogy az adott pozicíó üres-e vagy sem, ezek az adatok az értéktől függően hordozhatnak egyéb fontos információkat pl. egy színkódot vagy a kerethez tartozó elemeket.

# Műveletek:

* létrehozás, felszabadítás
* kiírás, ellenőrzés céljából
* bejárás
* sortörlés
* elem lekérdezése
* elem értékének megváltoztatása

# Létrehozás, felszabadítás

* Lefoglaljuk az adatszerkezet elemei számára szükséges helyet.
* Inicializáljuk az elemeket, pl. 0-ával, keret létrehozása, mely megkönnyíti a további műveletek alkalmazását, jelölhető pl. -1-el de bármilyen más a választott adattípusnak megfelelő értékkel, mely segítségével egyértelműen eldönthető, hogy az a kerethez tartozik.
* Ha a továbbiakban nincs szükségünk rá, akkor felszabadítjuk az előzőleg lefoglalt tárhelyet.
* Létrehozás: előfeltétel: -

utófeltétel: létrejön az adattároló, ebben az esetben a „Játéktábla”

* Felszabadítás: előfeltétel: -

utófeltétel:törlődik az adatszerkezet a memóriából

# Kiírás

* Kiírja a mátrixot az aktuális állapotában, célja ebben az esetben az ellenőrzés, hogy végrehajtódtak-e a meghívott utasítások.
* Közvetett kiírása: az elemei függvényében kirajzolja az adott színkódnak megfelelő, pixelek meghatározása érdekében felhasznált indexek segítségével, a konstans méretű négyzetet.
* előfeltétel: - (létezzen az adatszerkezet)
* utófeltétel: visszatéríti az aktuális állapotát a, illetve a grafikus ablakban a színes négyzeteket

# Bejárások

* Végigmegy a matrix elemei és közben vizsgálhat bizonyos feltételeket
* Ellenőrzések megvalósításánál, a bejárás közben vizsgálja, hogy az adott elem lemegy-e a játéktábláról vagy érintkezik-e más elemmel, stb.
* előfeltétel: - (létezzen az adatszerkezet)
* utófeltétel: megvizsgálta az elemeket

# Sortörlés

* Ha a játékosnak sikerült elérnie azt, hogy egy sorban minden üres helyre négyzet kerüljön, azt a sort törölje a mátrixból, illetve ez által a grafikus képernyőn sem fog megjelenni.
* előfeltétel: legalább egy betöltött sor
* utófeltétel: törlődnek a betöltött sorok

# Elem lekérdezése

* Egy bizonyos koordináta lekérdezése, az adatszerkezetben a hozzá tartozó elem értékének elérése. A mátrixban az üres helyek megtalálására almalazható.
* előfeltétel: - (létezzen az adatszerkezet)
* utófeltétel: visszatérít egy bizonyos értéket (implementációtól függően)

# Elem értékének megváltoztatása

* Egyéb tényezők és feltételek függvényében, ha módosítani kell egy értéket az adatszerkezetben, akkor ezt a műveletet alkalmazzuk.
* Ha a lefele mozgó alakzat elérte a végleges helyét, vagyis tovább nem tud mozogni, akkor a mátrixban megváltoztatjuk a koordinátáinka megfelelő értékeket.
* előfeltétel: esetleg ha van feltétel ami kell teljesüljön
* utófeltétel: megváltoztatja, “frissíti” az adott pozición lévő elemet

# Az adatszerkezet praktikussága a grafikus ábrázolásokhoz

Az játék kódja magába foglal grafikus könyvtárat is, mely lehetővé teszi az alakzatok, formák és színek kirajzolását a képernyőre. Maga a grafikus ablak is tekinthető mátrixnak, melyben a sorok száma az Oy tengelynek és az oszlopok száma az Ox tengelynek feleltethető meg.

Bővebben a felhasznált grafikus könyvtárról: [graphics.h](https://www.programmingsimplified.com/c/graphics.h) .