

# Tetris Programozói Dokumentáció

## Projekt felépítés

A projektben a mainen kívül 5 modul található + a megfelelő header file-ok, és van még egy extra Datas.h header file. A modulok "rétegesen" illeszkednek egymáshoz, vagyis a legtöbb modul működéséhez szükséges hogy bizonyos elemibb modulok include-olva legyenek. (pl: Move.c-hez szükséges a Placement.h).

Datas.h: A legelemibb modul. A legtöbb függvény futásához szükséges struktúrákat és makrókat tartalmazza.

Placement.h: A mozgó testek és a pálya interakciót tartalmazza.

Move.h: A testek mozgását tartalmazza.

Draw.h: Az SDL-lel kapcsolatos rajzolásokat és írásokat tartalmazza.

Game.h: A játékmechanikai működések tartalmazza.

Keys.h: A gombok lenyomására történő eseményeket tartalmazza.

Main.c: A játékhoz szükséges változók deklarálása és inicializálása, az SDL ablak inicializálása, a játék loop és a programkilépés előtt a megfelelő memóriaterületek felszabadítása.

## Modulok elemei

Datas.h:

Makrók: sizex: A pálya vízszintes mérete. sizey: A pálya függőleges mérete.

Struktúrák:

Block: Egy x és egy y koordinátát tartalmaz, ez reprezentál egy négyzetet a pályán.

Obj: Négy darab block-ot, és egy formakódot tartalmaz, ez reprezentál egy mozgatható objektumot a pályán.

Txt: A képernyőre íráshoz szükséges műveletek paramétereit tartalmazza.

Placement.h:

Működéséhez szükséges include-olni a Datas.h-t

Függvények:

Place: Szükséges bemenete a pálya (2d dinamikusan foglalt tömb) és egy block. A pályán elhelyezi a blockot amit egy 1-es reprezentál.(alapvetően a placeobj segédfüggvénye)

Dis: Szükséges bemenete a pálya és egy block. A blokkhoz tartozó x és y koordinátán a pályaértéket 0-ra változtatja.(alapvetően a removeobj segédfüggvénye)

Placeobj: Szükséges bemenete a pálya és egy obj. Elhelyezi a pályán az objektumot.

Removeobj: Szükséges bemenete a pálya és egy obj. Kiszedi a pályáról az objektumot.

Placepred: Szükséges bemenete a pálya és egy obj. Elhelyez egy predikciót a pályán.

Removepred: Szükséges bemenete a pálya és egy obj. Kiszedi egy predikciót a pályáról.

Genobj: Bemenete nincs, kimenete pedig egy obj. Egy 1 és 7 között generál egy formát, és ezt egy formától függő randomszámmal igazítja el, hogy a pálya felső két sorában hova kerüljön.

Move.h:

Működéséhez szükséges include-olni a Placement.h-t és az stdbool.h-t

Függvények:

Íránymozgáspárok: (downm,down,leftm,left,rightm,right)

Ezek a függvények párban vannak, egy-egy m-es segédfüggvény és egy elhelyezési ciklusfüggvény. A segédfüggvény bemenetébe csak egy obj kell, míg a ciklusfüggvényhez már a

pálya is kel, de mindkettő kimenete egy obj. A segédfüggvény csak elmozgatja az objektum koordinátáit a megfelelő irányba, a ciklusfüggvény pedig először kitörli az objektum állapotát a pályáról utána pedig elhelyezi az új pozícióját. Ezekből a párokból van 3 (jobbra, balra, le).

Esés és predikció: (dropm, drop, predm)

Az esés annyiban különbözik az iránymozgásoktól hogy itt a segédfüggvény bemenetére is kell a pálya mert itt a mozgatsnál számításba kell venni hogy a pályán belül emnyit lehet lefele menni.

A predikcióhoz pedig nincs szükség segédfüggvényre, mivel itt az aktuális objektumtól függ hogy hol van a predikcióhelyzete. Ennek megfelelően az akutális objektumot is rá kell helyezni a bemenetre, és a függvény egy predikciót helyez el a pályán.

Block mozgatasok: (ulm, urm, dlm, drm)

Ezek a függvények mind az objektum forgatásához szükséges segédfüggvények. Egy block a bemenetük és egy blockot adnak vissza. A függvényeknek a pályán való mozgatas irányának rövidítései (pl ulm- up left movement).

Forgatas: (shift, lrotate, lfr)

A shift egy segédfüggvény ami egy obj-t kér be és egy obj-t ad vissza. Vannak olyan formák amelyeknek kevesebb állása van mint a többinek, viszont mivel az objektumban a blockok mind el vannak nevezve és követve vannak ezért gyakorlatban minden formának 4 állása van. A shift kondició minimalizálás céljából megcseréli a megfelelő block párokat.

Az lrotate és az lfr az iránymozgásokhoz hasonló függvénpár. Az lrotate a formakód alapján elvégez az objektum koordinátáin egy bloklali irányú forgatást, az lfr pedig ezt elhelyezi a pályán.

Ütközés: (lcollide, rcollide, dcollide, rotcol)

Ezeknek a függvényeknek a bemenetén a pálya és egy obj van, kimenete pedig mindegyiknek egy bool. Eldöntik hogy a megfelelő mozgatasok elvégezhetők e vagy ütköznek e valamivel a pályán esetleg kívül esnek a pályán.

Draw.h:

Működéséhez szükséges a Datas.h include-olása.

Függvények:

Print: Bemenete egy karaktertömb, a képernyő (SDL\_Surface típusú pointer), és egy txt struktúra. Kirajzolja a beérkező szöveget a képernyőre a megfelelő karaktertípussal.

Map: Bemenete a képernyő. Megrajzolja a pályát.

Dmenu: Bemenete a képernyő és két txt. Megrajzolja a menüt.

Dblocks: Bemenete a képernyő és a pálya. Végig iterál a pályán és a megfelelő helyekre négyzeteket rajzol. Ez mutatja a felhasználó felé hogy a pályán mi van.

Dnext: Bemenete a képernyő és egy obj. A kis téglalapba a pálya mellett berajzolja az előre kalkulált következő elemet formakód alapján.

Settp: Bemenete egy txt két koordináta, egy szélesség és egy magasság. Beállítja a képernyőn lévő pozícióját az írandó szövegeknek, szinte minden írás művelet előtt szükséges.

Dpause: Bemenete a képernyő és egy txt. Megrajzolja a pause menüt.

Dcontrolhelp: Bemenete a képernyő, egy txt, egy gombszöveg koordináta, egy y koordináta, egy eseményszövegkoordináta, és két karaktertömb. A dcontrols segédfüggvénye. Beállítja a beérkező szövegeket a megfelelő pozícióba és kiírja.

Dcontrols: Bemenete a képernyő és három txt. A dcontrolhelp sgítéségével megrajzolja a Control menüpontot.

Dscore: Bemenete egy érték, egy karaktertömb, a képernyő, és egy txt. A játék futása alatt rajzolgatja a pontok értékét ha az megváltozik.

Dhighscore: Bemenete a képernyő, egy int pointer, egy karaktertömb, 3 txt és egy file pointer. Ez rajzolja ki a High Score menüpontot az elemntett maxpontszám alapján.

Game.h:

Működéséhez szükséges include-olni a Move.h-t és a Draw.h-t

Függvények:

Checkrow: Bemenete a pálya és a sortömb (1d int tömb). Végig iterál a pálya összes során és amelyikben 10 elhelyezett blokkot (azaz 2-est) talál ott a megfelelő sorindexet elmenti a sortömbben.

Checkloss: Bemenete a pálya, kimenete egy bool. Megnézi hogy a pálya második sorában van-e elhelyezett objektum, ha igen akkor igazat ad vissza.

Clrrow: Bemenete a pálya és a sortömb. Végig iterál a sortömbön és ahol 1-est talál ott a megfelelő indexen 0-ra állítja a pálya sorában található összes elemet.

Alldown: Bemenete a pálya és a sortömb. Végig iterál a sortömbön vonalon és ahol a sortömbben 1-est talál ott visszafelé el kezd iterálni a pályán és minden elhelyezett blokkot eggyel lemozgat.

Solidify: Bemenete a pálya és egy obj. Egy leérkezett objektumot elhelyez a pályán.

Zerow: Bemenete a sortömb. Végig iterál a sortömbön és 0-ra állítja az összes értékét.

Cntscore: Bemenete a sortömb kimenete egy int. Végig iterál a sortömbön és megszámolja hány egyes van benne. Ez alapján számol egy pontszámot amit visszaad.

Setspeed: Bemenete egy char pointer (sebesség) és a pontszám. A sebesség jelenlegi értéke és a pontszám alapján átállítja a sebességet.

Cleanfield: Bemenete a pálya. Végig iterál a teljes 2 dimenziós tömbön és az összes értéket 0-ra állítja.

Seths: Bemenete a pontszámra egy pointer és egy file pointer. Attól függően hogy az elért pontszám nagyobb-e mint a már elmentett legmagasabb pontszám átírja a legmagasabb pontszámot tartalmazó file-t.

Reachdown: Bemenete a képernyő, a pálya és 3 obj pointer. Elvégzi a szükséges függvényeket amikor az objektum leért a pálya aljára. Először is kiszedi a pályáról a predikciót, majd behelyez egy leérkezett objektumot az aktuális helyére. Ezután a következő értékét behelyezi az aktuálisba és a következőbe generál egy új objektumot. Utána újraszámolja a predikciót és rajzol.

Stdevent: Bemenete a képernyő, a pálya, a sortömb, a pontszám, egy karaktertömb, egy txt, és 2 obj pointer. Elvégzi a játékciklushoz szükséges műveleteket. Először megnézi a teli sorokat, aztán kiüríti őket. Ezután az elmentett sorindexek alapján leviszi a pálya tartalmát, áthelyezi a predikciót és számol egy pontszámot. Ezután kitisztítja a sortömböt és rajzol.

Checkfield: Bemenete a pálya, egy boolt ad vissza. Megnézi hogy van-e valami a pályán ha igen akkor igazat ad vissza.

Keys.h:

Működéséhez szükséges includeolni a Game.h-t. Sokbemenetes függvények vannak benne legtöbbje arra szolgál hogy kompaktan tartsa a main-t.

Függvények:

Kdowng: Bemenete a képernyő, a pálya és 2 obj pointer. A játékon belüli lefele gombnyomásra történő eseményeket hajtja végre. Lemozgatja az objektumot, számol egy új predikciót és rajzol.

Kdownp: Bemenete a képernyő és a kurzorra egy pointer. A pause menü alatt megnyomott lefelegombra történő eseményeket hajtja végre. Átállítja a kurzort és rajzol.

Kdownm: Bemenete a képernyő és a kurzorra egy pointer. A menü alatt megnyomott lefelegombra történő eseményeket hajtja végre. Átállítja a kurzort és rajzol.

Kleftg: Bemenete a képernyő, a pálya és 2 obj pointer. A játékon belüli bal gombnyomásra történő eseményeket hajtja végre. Balra mozgatja az objektumot, számol egy új predikciót és rajzol.

Krightg: Bemenete a képernyő, a pálya és 2 obj pointer. A játékon belüli jobb gombnyomásra történő eseményeket hajtja végre. Jobbra mozgatja az objektumot, számol egy új predikciót és rajzol.

Kupg: Bemenete a képernyő, a pálya és 2 obj pointer. A játékon belüli fel gombnyomásra történő eseményeket hajtja végre. Forgatja az objektumot, számol egy új predikciót és rajzol.

Kupp: Bemenete a képernyő és a kurzorra egy pointer. A pause menü alatt megnyomott felgombra történő eseményeket hajtja végre. Átállítja a kurzort és rajzol.

Kupm: Bemenete a képernyő és a kurzorra egy pointer. A menü alatt megnyomott felgombra történő eseményeket hajtja végre. Átállítja a kurzort és rajzol.

Kenterg: Bemenete a képernyő, a pálya, a kurzor, a pontszám, 2 boolpointer, 3 txtpointer, egy filepointer és 3 obj pointer. A játékon belül a pause menüben lenyomott enter gombra történő eseményeket hajtja végre. Ha a save game van kiválasztva (kurzor 0), akkor a játék helyzetét kiírja a savegame.txt file-ba és kiírja a képernyőre hogy a játékot elmentette. Ha az Exit to menu van

kiválasztva (kurzor 1), akkor leállítja a játékot, kirajzolja a menüt, alaphelyzetre állítja a kurzort és kitakarítja a pályatömböt.

Kenterg: Bemenete a képernyő, a pálya, a kurzor, a pontszám, 4 boolpointer, 4 txtpointer, 2 filepointer és 3 obj pointer, egy karaktertömb. A menün belül lenyomott enter gombra történő eseményeket hajtja végre. Ha a Start game van kiválasztva (kurzor 0), akkor elindítja a játékot, megrajzolja a pályát, kiírja a pontszámot, legenerálja az aktuális és következő objektumokat, számol egy predikciót, elhelyezi az objektumokat a pályára és rajzol. Ha a Load Game van kiválasztva (kurzor 1), akkor megnyitja a savegema.txt-t betölti az értékeket, elindítja a játékot, megrajzolja a pályát és a pontszámot, számol egy predikciót, azt elhelyezi és végül rajzol. Ha a Controls van kiválasztva (kurzor 2), akkor megrajzolja és elindítja a Control menüpontot, a kurzort pedig alaphelyzetbe állítja. Ha a High Score van kiválasztva (kurzor 3), akkor megrajzolja és elindítja a High Score menüpontot, a kurzort pedig alaphelyzetbe állítja. Ha az Exit Game van kiválasztva (kurzor 4), akkor leállítja a programot.

#### Main.c:

A main include-olja a time.h-t, stdio.h-t, stdlib.h-t és a Keys.h-t amin keresztül az összes többi modult. Itt található a timer függvénye és maga a main függvény.

##### Init:

A függvény elején létrehozza az összes szükséges változót és lefoglalja a memóriát a pályának. Meghívja az SDL ablakot és az időzítőt. Utána inicializálja a szövegeket a megfelelő betűméretekkel és színekkel. Innentől megrajzolja a menüt, beállítja a timert és elindítja a játék loopot.

##### Loop:

A loopon belül a különböző állapotokat boolok jelzik. Ezek alapján dönti el hogy mikor mit csinálhat, mikor melyik gombot lehet megnyomni és az mit csinál. A legtöbb gombra van export függvény a Keys.h-ba, de néhány helyen fölöslegesnek ítélttem. A useventen belül csak akkor hajtódnak végre az utasítások ha a játék éppen történik és nincsen leállítva (pause). Mivel a timer maga az konstans a loopra nézve ezért a loopba beépítettem egy tick-et ami folyamatosan a számol és a usevent csak mind a speed változóval való megfelelő leosztásra hajtódik végre. Ezért minél kisebb a speed változó annál gyorsabb a játék. Az event először megnézi hogy el lett e veszítve a játék és ha igen, akkor végre hajtja a megfelelő kilépési procedúrát. Ennek az ellenőrzése után a különböző feltételeknek megfelelően futtatja a játékot. A ticken kívül pedig le van kezelve az elvesztés esetén a már kilépett játékban történő esemény.

##### Quit:

Ha a quit bool igazra lett állítva és kiléptünk a loopból, a program még felszabadítja a memóriát és csak utána fejeződik be.