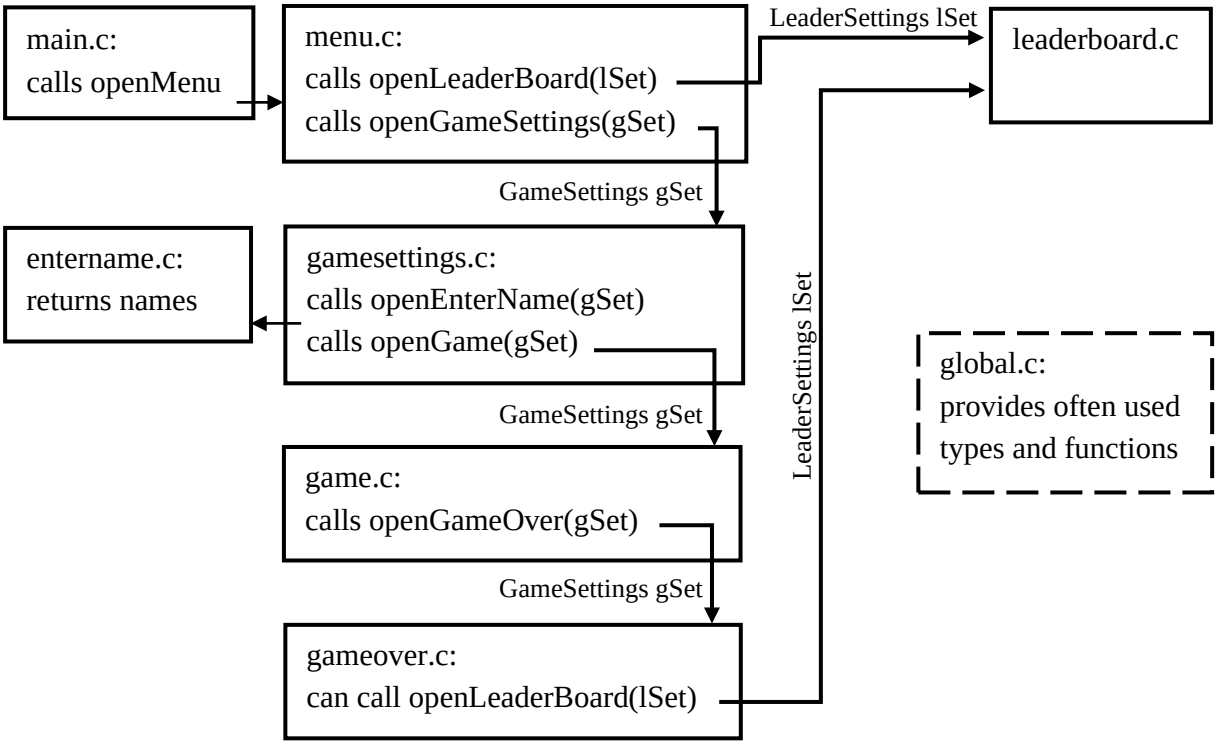
Piller Trisztán WHKJZX

Lil'Snek programozói dokumentáció

KÖRNYEZET: CodeBlocks 17.12 IDE.

FORDÍTÁSHOZ SZÜKSÉGES: InfoC-SDL2.0 grafikus könyvtár + művészi megjelenéshez „data" mappa és tartalma.

FELÉPÍTÉS: A „data" mappában találhatóak „png" almappában a szükséges képi elemek, a „music" almappában a zenei fájlok, valamint a főmappában a globálisan használt betűtípus és a játékmentéseket tartalmazó szövegfájl. Az egyes modulok, és azok fő függvényei felelnek külön-külön egy-egy ablak megjelenítéséért, valamint a részfeladat megvalósításáért. Hívási gráf:



MAIN.C: Fő programmodul, beinicializálja a grafikus könyvtárat (SDL), a betűtípus könyvtárát (TTF), a zenei könyvtárat (Mix), elindítja a játékprogram első részfeladatot, a főmenüt. Visszatéréskor biztosítja a könyvtárak teljes bezárását.

GLOBAL.C: Az összes almodul számára biztosítja a gyakorta használt funkciókat (gombok megjelenítése, felirat megjelenítése, nyomott gomb felismerése). Összegyűjti a globálisan használt típusokat:

- GameSettings: egyszerű struktúra, amely a játékbeállításokat passzolja tovább modulok közt.
- LeaderSettings: hasonlóképp, a kívánt beállításokat adja tovább megjelenítésre.
- WindowSpecs: az éppen használatban lévő ablak jellemzőit sűrítő struktúra. Segítségképp a GameSettings-re tartalmaz egy mutatót.
- Button, ButtonSet: gombkoordinátákat, nevet tartalmazó rekordtömbtípus és a halmazt jellemző struktúra.

MENU.C: Egyetlen publikus függvénye – openMenu() – biztosítja a menü összes funkciójának megvalósítását, benne eseményekként kezelve a gombnyomásokat. Privát függvények röviden:

- sdlInit() és sdlClose(): az ablak, renderer, ikon és egyéb grafikus/zenei komponensek inicilializálását végzi (ezek minden almodulban jelen vannak).
- initGameSettings() és initLeaderSettings(): inicializálja a struktúrákat.
- initStartingLook(): ablak nyitásakor felel a felhasználót köszöntő elemekért.
- drawRightPanel(): statikus jobb panel grafikus elemeit kialakító függvény.
- returnHere(): másik modulból való visszatéréskor biztosítja az ablak újranyitását.

GAMESETTINGS.C: Beállíthatóak a játszma részletei (lásd specifikáció).

- showLabels(), showMode(), showGridXY(), showArrows(), showNames(): képernyő grafikus elemeit irányító, átrajzoló függvények.
- initStartingLook(): a megnyitott ablak elemeit megjelenítő függvény.

ENTERNAME.C: A nevek begépelését megvalósító modul. (SDL-ben körülményes egy input box beillesztése.)

- inputText(): legálisan beszerzett függvény https://infoc.eet.bme.hu/sdl/#7
- enterName1(), enterName2(): 1-es és 2-es játékos nevének megadására lehetőséget kivitelező függvények.

GAME.C: Elindul benne a játszma, a játék lényegét adó modul. Időzítésre mennek végbe lépést generáló események, updateGame() foglalja össze egy lépés működését.

- GridType, gridmapping1/2, getCoordOfGridType(): A bitmapból (map.png) való kiolvasáshoz megadott konstansokat tartalmazó enumerátor, x-y koordinátatömb és függvény. Így habár nem hatékony, de későbbi módosításhoz könnyen átlátható a kód.
- Direction, SnakeCell, Player: ezen struktúrák jellemzik a kígyót, annak irányát, következő állapotát, láncolt listáját az elemeknek.
- Bunny: az elhelyezett zsákmány rácskoordinátáit gyűjtő struktúra.
- timerCb(): callback függvény az SDL időzítőjéhez.
- copyToGrid(): kígyó mozgása alatt a testrészeket megfelelő módon kirajzoló függvény.
- initGame(), initBunny(), initPlayer(), destroyPlayer(): „constructor” a játéknak, a zsákmánynak, a kígyónak és free-elő függvény a kígyó láncolt listája számára.
- initGridSystem(): az alap rácsrendszer kirajzolása a játék elején.
- showPoint(): kimutatja a játékos pontszámát játék közben.
- checkBunny(), putNewBunny(), isFree(), growPlayer(), movePlayer(), checkCrash(): mind a játék főmüködését szolgáló függvények, updatePlayer()-ben hívódnak meg alapvetően. Sorrendben: első megvizsgálja, következő lépésben rajta lesz-e a zsákmányon, azaz befalja-e a kígyó. A második új nyulat próbál letenni, ha isFree() úgy határozza, hogy oda tehet új nyulat. A negyedik növeli a kígyót listaelem-beszúrással úgy, hogy megállítja a kígyót, csak a fej mozog, és új testelemet tesz be a fej mögé. Az ötödik mint egy állapotgép shifteli tovább az irányokat. „Player” struktúra „state” változója határozza meg, mit kell betolni a fej „dir” (irány) változójába, és a fejtől kezdve minden testelem iránya továbbshiftelődik a következő elemre. Így mozgatja a kígyót. A hatodik megnézi, ütközés történik-e a lépés során, ez határozza meg updatePlayer() visszatérési értékét, azaz hogy game over (játék vége) van-e.

GAMEOVER.C: Játszma lefutása végén meghívódó modul, amely biztosítja a mentést és az eredmény utáni interakciót.

- showTitle(), showFinalPoint(), initStartingLook(): megjelenítést szolgáló függvények.
- saveGame(): alapvetően a specifikációban közölt módszerrel szövegfájlba menti a játszma eredményét, viszont eltér annyiban, hogy a szükségtelen játékmező szélességet és magasságot kiiktatja.

LEADERBOARD.C: 4 féle kimutatási módszerrel dolgozik aszerint, többjátékos vagy egyjátékos; easy vagy challenge módban kéri a felhasználó a kimutatásokat.

- initStartingLook(): kezdeti megjelenítés
- loadFile(), insertResult(), freeResult(): Az első függvény beolvassa a fájl sorait, lényegeseket listába menti insertResult által. freeResult() biztosítja a felszabadítást.
- showResults(), showPagingButtons(): a koncepció lényege, hogy egy oldalon 10 eredményt lehessen megjeleníteni, eszerint írja ki a függvény a listából az eredményeket. A másik függvény ügyel a megfelelő gombok megjelenítésére, kezelésére.

A függvények előtt a programkódban találhatóak rövid leírások, amelyek segítik megértetni a visszatérések célját, a függvénynevek, a paraméterek segítik a megértést, néhol szükséges az SDL-es típusok ismerete (pl.: SDL_Rect egy téglalapot jellemző struktúra x, y, w (szélesség), h (magasság)).