

Aknakereső dokumentáció

Felhasználói dokumentáció

A program induláskor kezdő módot indít nekünk. A pályán tett első indítás elindítja a számlálót. Ekkor indul a játék. A játékot akkor nyerjük meg, amikor a pálya összes aknáját megtaláljuk anélkül, hogy felfednénk őket. Ha aknát fedünk fel a játékot elvesztettük. A játék elvesztésénél a program felfedi az összes többi akna pozícióját is. A felfedett mezők mindig a környezetükben lévő akna számát mutatják meg. Ha nincs akna a mező környékén, akkor a környező területeket is feloldja addig, amíg nem számmal vagy a pálya határával szegélyezett terület alakul ki.

A műszerfalon 3 dolog van segítségünkre:

Bal oldalon a zászlószámláló található, ami eredetileg a pályán elhelyezett akna számával egyenlő. De a lehető zászlók száma nem korlátozott. A számlálót tehát negatívba is vihetjük. Zászlót úgy tehetünk le, ha egy mezőre jobb klikkelünk. Ekkor nem tudjuk feloldani a mezőt. Ha jobb klikkelünk egy zászlóra, a mezőn kérdőjel jelenik meg. Ekkor már feloldható. Ha egy kérdőjelre kattintunk jobb egérgombbal, akkor üressé válik a mező. Ha eltüntetünk egy zászlót a zászlószámláló növekszik egyel.

Középen a smiley gomb van. Alapvetően mosolyog. Amíg nyomva tartjuk az egeret aggódik, ha aknát fedünk fel halottá válik, ha megnyerjük a játékot, napszemüveget kap. Lekattintása újraindítja a játékot.

A jobb oldalon van a számláló. Az első mező felfedésétől a játék végéig számol.

A menüről:

Az első menüpont a fejlécben a Game.

Ennek kattintásakor a következő lehetőségeket látjuk:

- New game: ugyanúgy új játékot indít, mint a smiley gomb.
- Beginner: kezdő nehézségű játék indítása
- Intermediate: haladó nehézségű játék indítása
- Expert: szakértő nehézségi szintű játék indítása
- Custom: Egyedi szint. Ekkor egymás után megjelenő mezőkben adhatjuk meg a pálya adatait:
 - A pálya szélessége legalább 9, maximum 24.
 - A pálya magassága legalább 9, legfeljebb 30.
 - A pályán elszórható akna száma 10-300-ig terjed, de a pályán lévő mezők számától 1-el kevesebbnek kell lennie.
 - **Enter megnyomásával tudunk adatot bevinni és továbbugrani.**
- A Leaderboard pontban megnézhetjük az időeredményeket. Ha Anonymous 999 jelenik meg akkor az gyári adat. Még nem írta felül játékos. **A kilépéshez bárhova kattinthatunk a program felületén.**
- Az utolsó a kilépés. Ez bezárja a programot.

A második menüpont a játék:

Ekkor a játék alapvető rövid leírása és a fejlesztéssel kapcsolatos minimális információ látható.

A kilépéshez kattintson bárhová a felületen.

A menübe lépés és kilépés minden esetben a játék megszakításával jár. Azaz visszalépéskor új pálya vár minket.

A dicsőségtábláról:

Ha megnyerjük a játékot, és az eredményünk kisebb, mint az előző legjobb, akkor bekerülünk a dicsőség listába. Ekkor a program bekéri a legfeljebb 10 karakter hosszú nevünket. A dicsőségtábla háttérfájljába ez bekerül az időeredményünkkel, de a dicsőségtáblában is rögtön látható lesz.

A tábla csak a legjobb eredményt rögzíti szintenként. Az egyedi szint nem tartozik bele ebbe a listába, mivel sok különböző nehézségű pálya készíthető el benne.

Programozói dokumentáció

A program működése:

Alapvető változók és funkcióik:

A program az első lépésként az ablakot jeleníti meg, deklarálja a játékhoz szükséges változókat, beállítja a könnyű szint alapértékeire és meghívja az `uj_jatek_general` függvényt. Ez a függvény az ablak méretét, hátterét meghatározza a kapott adatok alapján, továbbá legenerálja a fejléc menüpontjait, a műszerfal elemeit és magát a pályát. Természetesen ehhez a művelethez a `main` függvényben tárolt pályatérkép és feloldó mátrixokat felülírja. A működéshez szükséges ismerni a változókat és funkcióikat:

Textúrákat tároló mutatókból álló tömb: a dinamikusán elmentett textúratömb a pályatérkép mátrix elemeivel közvetlenül indexelhető. Ezáltal a feloldás gyors és műveletmentes folyamat.

Pályák struktúra: tartalmazza a pálya szélességét, magasságát, aknaszámát. Ehhez tartozik a pályakód egész változó, amely ezt indexeli, így könnyen tudjuk megállapítani, hogy melyik pálya adatait kell használnunk és melyek azok.

A pályatérkép egy olyan mátrix, amely tartalmazza a pálya teljes elrendezését. Pontosan és egyértelműen tartalmazza a mezők értékét. 0-8, a mező környezetében található akna szám, 9 ha ő maga az akna.

A pálya vezérlését egy feloldva nevű mátrix segíti. Értékei a következőképp változnak: 0, ha a mező érintetlen; 1, ha zászló van rajta; 2, ha kérdőjel van rajta; 3, ha feloldottuk. Ebből kitalálható, hogy ez a mátrix a felhasználó nyomon követésére nyújt takarékos és átlátható megoldást. A mátrix abban is segít, hogy ne oldjunk fel újra olyan mezőt, amit már feloldottunk tehát mellőzzük a felesleges lépéseket.

Két logikai változót is tárol a program: az egyik a játék végét jelöli. Ha szünetel a játék, akkor nem fogunk a pálya pozíciójában tett kattintások lekezelésével foglalkozni. Ez menübelépéskor is igaz természetesen. A másik logikai változó a játék kezdetét jelzi. Ugyanis az első mező feloldása jelenti a játék kezdetét, ekkor indul az óra. Az óráról részletes kifejtés lentebb található.

Két vezérlő egész szám változó is van. Az első a feloldások számát jelöli. Ez azért fontos, mert ha mindent feloldottunk már, ami nem akna, akkor nyertesnek minősül a játék. Ezután már láthatjuk az összefüggést az előző bekezdés változóival. A másik a mód. Szintén fontos az eseményvezérlés szempontjából. 3 fajta módunk van: játék, 1. menütípus, 2. menütípus. Ezeket 0-2-ig jelölik számok. A mód értéke alapján döntjük el, hogy miként kezeljük az eseményeket.

Ez utóbbi 4 bekezdés változói az egyszerűség és az egyszerűsített paraméterezés érdekében egy struktúrában vannak.

Eseményvezérlés:

A vezérléshez egy struktúrát használunk, amely tartalmazza az egér lenyomásának pozícióját és egy logikai változóval azt is, hogy melyik gomb lett lenyomva. A felengedésnél ezáltal állapítjuk meg a klikket. Ha minden adat azonos a lenyomással, akkor klikk esemény történt. A lenyomás a felengedésig mást is von viszont magával. Játék közben a pálya területén kattintva a műszerfalon található smiley gomb ekkor aggódóra vált. Természetesen textúra cserélgetés történik ilyenkor és csak bal klikkre játékmódban működik.

A jobb klikk eseményei: a jobb klikk csak játékmódban használható. A jobb_klikk_esemenykezelő meghívása történik. Szüksége van a pálya adataira, tehát a pályák pályakódadik elemére, a textúrákra a klikk x és y pozíciójára és feloldva mátrixra. Megállapítja, hogy a terület melyik mezőjén van a klikk, ezáltal indexelni tudja a feloldva tömböt. Ha az aktuális érték 0, akkor 1-re vált és felfesti a mező pozíciójára a zászló textúrát, ha 1, akkor 2-re vált és felfesti a kérdőjelet. Viszont a másodikról 0-ra fog visszaugrani. Ha már fel van oldva a mező, akkor nem végez rajta műveletet. Továbbá a programban van egy funkció miszerint van a műszerfalon egy zászlószámlálónk. Az akna számáról indul, de jobb klikkelgetés hatására negatívba is eshet az értéke. Ennek a számlálónak a működését is ez a függvény biztosítja néhány segédfüggvény használatával. A segédfüggvényekről részletes leírás lejjebb található.

A bal klikk eseménykezelés jóval összetettebb.

Először is teszteljük a módot. Ha játékmódban vagyunk, tehát a mód változó értéke 0, akkor tesztelünk a fejléc menüpontjaira. Ezek kattintásakor a jatek_menu_general vagy a help_ful_general függvényt hívjuk meg. Ezek felfestik a kijelzőre az adott menü feliratait. Továbbá a smiley gomb pozíciójára is tesztelünk, amely funkciója az új játék indítása. Ekkor a tisztítás függvény a változókat alapértékre állítja és felszabadítja a dinamikusan foglalt területeket, és meghívjuk az uj_jatek_general függvényt.

Azonban arra is tesztelünk ha a pályára kattintottunk. Ennek ellenőrzését külön logikai visszatérési értékű segédfüggvény végzi. Itt történik meg a játék vége logikai ellenőrzés és az első kattintás tesztelése is, azaz a timer indítása. Amennyiben átesünk ezeken a teszteken a palyan_bal_klikk kerül meghívásra. Hasonlóan működik mint a jobb klikk eseménykezelője. Ahol zászló van az nem feloldható, így ha az aktuális mező feloldási értéke 1, akkor visszatér 0-val. Ellenben ha pályatérkép aktuális eleme 0, akkor egy rekurzív függvényt hív meg. Ez a felfedő. Felfedi a lehető legnagyobb nem nulla határú területet. Visszatér a feloldott mezők számával. Ekkor ezt az értéket küldi vissza a hívó is. Ha a pályatérkép aktuális értéke 9, akkor -1-et küld vissza és feloldja az összes aknát, a kattintott mezőre a piros háttérű aknát teszi. Végül, ha ezen esetek egyike sem teljesül, csak az aktuális mező oldódik fel és visszatér a

függvény 1-el. A hívó a függvény értékét hozzáadja a feloldások számához, kivéve ha -1 tér vissza, mert akkor játék vége. Fordul az ezt jelző logikai értéke és a smiley textúrát vált. Ekkor a számláló is megáll. Ha viszont a visszatérésnél az adott játéknézés LboardAdat struktúrában tárolt eleménél kisebb az óra adatait tároló struktúra másodperc mérője, akkor az Leaderboard_kezelot hívjuk meg. Azaz rögzítésre kerül az új eredmény név és idő alapján. Ezután a szokásos új játék rutint hívjuk meg.

Ha a mód értéke 1 akkor külön teszteljük a menü feliratok pozíciójára a klikk helyét. Az első 4 csak az uj_jatek logikai és a pályakód változók értékét módosítja. Az 5. az egyedi játékmódot választja ki. Ekkor történik meg az adatbekérés az egyedi_adat_bekero függvénnyel. A függvény az input_text függvényt használja és egészszé konvertálja a kapott értéket, majd visszaküldi. Viszont határértéket is meg kell adni neki, amelyek között elfogadható az adott szám. Amíg a beolvasás nem zajlik le, nem lehet az X-szel bezárni az ablakot. Ha az adat megfelelő kilép a függvény a határértékeket tesztelő while-ból és visszatér. Ez a folyamat 3-szor zajlik le, amíg a 3 külön adatát be nem kértük a pályának. Ekkor pályák struktúra utolsó elemét írjuk felül. Feliratot is megjelenítünk a mező felett, hogy a felhasználó tudja, mit adjon meg. Miután végeztünk a bekéréssel az új játék és a pályakód változói felülíródnak. A következő menüpont a dicsőségtábla kiírása. A dicsőségtáblát a standard módon beolvasott és kiírt LboardAdat struktúra tárolja. A kiírás for ciklussal történik magától értetődő módon. Az utolsó a kilépés gomb, amely kitisztítja a memóriát és bezárja az SDL ablakot. Ekkor a main is visszatér 0-val. A tesztelés végén megnézzük az új játék változó értékét és ha igaz, akkor új játékot indítunk. Ekkor a már leírt függvénnyel felülírjuk a felületet és a változókat.

A 3. mód a legegyszerűbb. Ez a help menü és a dicsőségtábla esetén használandó. Ekkor nem nézzük a klikk pozícióját, mindenképp kilépünk új játék készítésével.

A kód felépítése és egyéb függvények:

Amelyek elmagyarázása szükséges volt az előző egységben azokról itt nincs részletes leírás.

Modulok és függvényeik:

1. esemenyvezerles
 - a. Uint32 idozit
 - i. Ez az SDL_AddTimer által hívott függvény. Feladata a szöveg kirajzolása a kapott struktúra alapján. Ez az adatszerkezet tartalmazza a pálya szálességét, amiből a pozíciót számolja és kiírja oda a másodpercek számát, ami szintén benne van az SDL_Renderer mellett a struktúrában.
 - b. uj_jatek_general
 - i. Már volt róla szó korábban. A vezérlő változók értékét alapállába helyezi és generál új játékteret.
 - c. palyan_bal_klikk
 - d. jobb_klikk_esemenykezelo
 - e. felirat_general
 - i. Az összes eddig említett felületen megjelenő felirat megjelenítéséért ez a függvény felelős. A pozíció a sztring és a betűméret alapján kirajzolja a szöveget
 - ii. Egyéb opció, hogy a szövegnek lehet egy széles háttere, amellyel elfedi az előző feliratot. Ez jól jön a számlálónál.

- f. felfedo
 - i. Rekurzív. Amennyiben minden környezetében lévő mezőre meghívja önmagát, amennyiben 0 a pályatérkép aktuális értéke. Nem hívja meg magát olyan mezőre sem, amelyet már feloldottunk. Számolja a feloldások számát.
- 2. fjlkezeles
 - a. Leaderboard_kezelo
 - i. a felületen létrehozza a bekérő felületet. Legfeljebb 10 karakter hosszú szöveg felel meg. A nev_bekero függvényt használja. Közvetlenül felülírja a LboardAdat struktúra tömb pályakód szerint indexelt értékét, amelyet paraméterként kap meg.
 - ii. Meghívja a kiir függvényt, hogy azzal felülírja a fájlt .
 - b. Beolvas és kiir: Hagyományos módon beolvasó és kiíró függvények
- 3. main
 - a. main
 - b. tisztitas
 - i. Az új játék meghívása előtt ez szabadítja fel az előző pálya dinamikus memóriaterületeit és állít alapértéket a változókon
 - c. sdl_init
- 4. menu
 - a. jatek_menu_general
 - b. Lboard_mutato
 - i. A teljes dicsőségtáblát kiírja.
 - c. egyedi_adat_bekero
 - i. Alsó és felső határ közötti int értékek beolvasására használatos. Az input_text függvényre támaszkodik. A felületet teljesen felülírja.
 - d. nev_bekero
 - i. Alsó és felső határ közötti int értékek beolvasására használatos. Az input_text függvényre támaszkodik. A felületet teljesen felülírja.
 - e. input_text
- 5. palyageneralas
 - a. textura_beolvasas
 - i. For ciklussal olvassa be a textúrákat
 - b. feloldas_terkep_general
 - i. A feloldva mátrixot generálja és a rámutató pointert küldi vissza
 - c. palya_rajzolo
 - i. az uj_jatek_general hívja meg. Ez a függvény rajzolja fel a pálya mezőit
 - d. palya_general
 - i. A pályatérképet generálja. Elsőként lefoglalja a területet és 0-val feltölti. Ezután random pozíciókba elszór aknákat. Ha letesz egy aknát, akkor a környező elemek értékét növeli 1-el, figyelve arra, hogy ne fusson ki a mátrix határain kívülre és másik akna értékét ne növelje. A mátrix pointerével tér vissza.
- 6. segedfuggvenyek
 - a. Xkoordinata
 - i. A kattintás x ponja alapján a pálya x koordinátáját számítja ki.
 - b. Ykoordinata

- i. A kattintás y ponja alapján a pálya y koordinátáját számítja ki.
- c. zaszloszamlalo
 - i. Megszámolja a lerakott zászlók számát a pályán. Ezt a feloldva mátrix és a pálya struktúra aktuális eleme alapján teszi
- d. palyan_van
 - i. Megvizsgálja, hogy a kattintás a pálya területét érte-e.
- e. Konvertal_sztringge
 - i. Egy pointert küld vissza, tehát figyelni kell a memória kezelésre. Konvertálja előjeles sztringé a kapott számot. Ez kell a számlálónak, hogy kiírja a számlálás aktuális értékét.
- f. Konvertal_egessze
 - i. kapott sztringet számmá konvertál és visszaküldi. 0 tér vissza akkor is, ha a formátum hibás.