

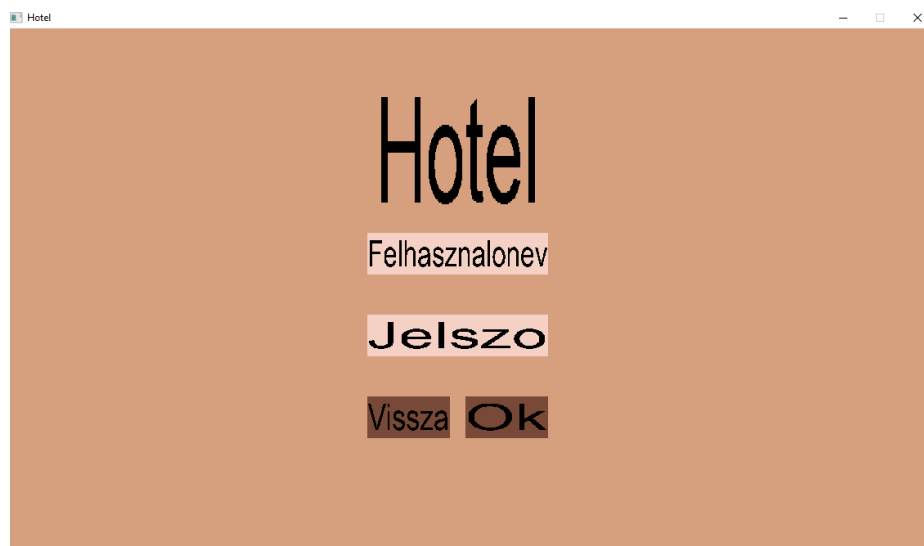
# Nagy házi programozói dokumentáció

## Hotelszobák (félkész)

### A program jelenlegi verziója

A program jelenleg félkész állapotban van. Képes megjeleníteni **mindkét menüt**, tehát a Login, és a Felhasználói ablakot. A Login ablak már teljes mértékben funkcionál, vagyis **működnek a gombok, beviteli mezők**, valamint képes megkülönböztetni az első elindítást a többitől, ezért a **felhasználó már tud regisztrálni, illetve belépni** a Felhasználói ablakba. A Felhasználói ablakon belül, a program képes **kirajzolni minden menü elemet**, valamint a Hotel alapértelmezett **szűrői**(férőhely, klíma, terasz) által lehet válogatni a hotelszobákat.

A Login ablak futásidőben:



Hotel

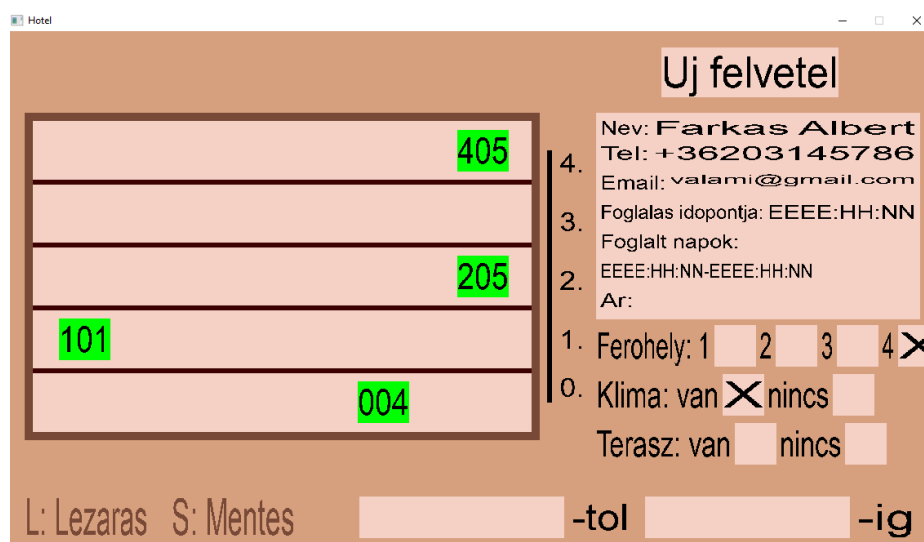
Hotel

Felhasznalonev

Jelszo

Vissza Ok

A Felhasználói ablak futásidőben, szűrővel beállítva:



Hotel

Uj felvetel

4.				405
3.				205
2.	101			
0.				004

Nev: Farkas Albert  
Tel: +36203145786  
Email: valami@gmail.com  
Foglalas idopontja: EEEE:HH:NN  
Foglalt napok: EEEE:HH:NN-EEEE:HH:NN  
Ar:

1. Ferohely: 1 ☐ 2 ☐ 3 ☐ 4 ☒

0. Klima: van ☒ nincs ☐

Terasz: van ☐ nincs ☐

L: Lezaras S: Mentos  -tol  -ig

## A program felépítése és modulok

A program jelenleg **5 modullal** dolgozik, ezek: main.c, megjelenites.c, fajladatkezeles.c, eventek.c, bejelentkezesmenu.c. A main.c tartalmazza a program **fő loopját**, valamint azokat a változókat, struktúrákat, amelyek **elengedhetetlenek a két ablak működéséhez**. SDL2 program lévén a megjelenites.c tartalmazza a **grafikus függvényeket**, illetve az egyes grafikus alakzatok, kezelő felületek, gombok, beviteli mezők **kezdőértékekkel való ellátását**. A fajladatkezeles.c fájlban találhatóak a **fájlba írás és olvasás műveletek**, valamint az **adatstruktúrák deklarációi**, jelenlegi verzióban, ez az Ár struktúrán figyelhető meg. Az eventek.c fontos modul, mivel ez tartalmazza a főloopen **belüli loopot**, mely a felhasználóval tartja a kapcsolatot, vagyis a **felhasználó bemeneteket**(egér kattintás, gépelés) **kezeli**, s dönti el ezáltal, hogy milyen **művelet következzen**. A bejelentkezesmenu.c tartalmazza a Login ablak **függvényeit**, ezek kerülnek meghívásra kattintás, gombnyomás esetén.

## Könyvtárak

A program használja az <stdio.h> könyvtárat, ezen kívül a **logikai függvények** miatt beépítésre került az <stdbool> könyvtár is, a bonyolultabb **matematikai számítások** miatt bekerült a <math.h> könyvtár. A sok **sztring művelet** miatt a <string.h> is használatban van. A **memória kezelés** elengedhetetlen, ezért a program használja az <stdlib.h> valamint a „debugmalloc.h” memóriakezelő könyvtárakat. Az SDL2 **megjelenítés, és a felhasználói bemenet olvasása** miatt a program tartalmazza az <SDL.h>, <SDL2\_gfxPrimitives.h>, <SDL\_ttf.h> könyvtárakat.

## Adatszerkezetek és struktúrák

A program jelenlegi verziója még nem tartalmazza a foglalási adatok adatszerkezetét.

```
1. typedef struct ar
2. {
3.     int arak[6];
4. }ar;
```

Az ár adatszerkezet a fajladatkezeles.c modulban található, és **egy 6 elemű int tömböt** tartalmaz, itt vannak eltárolva az árak, a hotelszobák szolgáltatásai szerint, első elindításkor ez a struktúra **alapértelmezett értékekkel** töltődik fel.

```
1. typedef struct gomb
2. {
3.     int kezdposX, kezdposY, vegposX, vegposY;
4.     int nevkezdposX, nevkezdposY, nevvegposX, nevvegposY;
5.     char nev[21];
6.     bool aktiv;
7.     bool kivlasztva;
8.     int r, g, b;
9. }gomb;
10.
11. typedef struct alakzat
12. {
13.     int kezdposX, kezdposY, vegposX, vegposY;
14.     int r, g, b;
15. }alakzat;
16.
17. typedef struct szo
```

```

18. {
19.     int kezdposX, kezdposY, hossz, szelesseg;
20.     char nev[21];
21.     int r, g, b;
22. }szo;

```

A megjelenites.c modulban található három struktúra, amelyek a három fő megjelenítendő típus adatait tartalmazzák. A gomb struktúra először tartalmaz 4 int változót, amik a téglalap alakú gombok **négy sarkának koordinátáit** tartalmazzák. Ezenfelül a következő 4 int változó a **szövegdoboz 4 sarkának koordinátáit** tartalmazzák, ez azért szükséges, mert vannak olyan gombok, melyek egyben beviteli mezők is, ezért gépelésre a **szövegdoboznak változó méretűnek** kell lennie. Ezenfelül tartalmaznak még egy **név tömböt**, ami ebbe van, az kerül megjelenítésre a képernyőn, kettő bool változót, az első azt határozza meg, hogy **kattintható legyen e a gomb**, a második pedig, hogy **meg legyen e jelenítve a képernyőn**. Az utolsó három változó a gomb **színét** tárolja. Az alakzat hasonló mint a gomb, de statikus, csupán dizájn elem. A szó alakzat azért van, mert a menü tartalmaz olyan szavakat, amelyek nem gombok szavai.

```

1. typedef struct hotelszobak
2. {
3.     int ferohely;
4.     int klima;
5.     int terasz;
6. }hotelszobak;

```

A hotelszobak struktúrát az eventek.c modul tartalmazza, ez azért szükséges, mert ilyen struktúrában készül el **alapértelmezetten a hotel felépítése**, vagyis, hogy az egyes hotelszobák milyen szolgáltatásokat tartalmazznak. 2-es jelöli ha van, 1-es ha nincs az adott szolgáltatás, a férőhelynél a helyek száma van.

```

1. typedef struct FelhasznaloAdatok
2. {
3.     char felhasznalonev[21];
4.     char jelszo[21];
5. }FelhasznaloAdatok;

```

A bejelentkezesmenu.c tartalmazza a FelhasznaloAdatok struktúrát, egy ilyen struktúrába kerül **beolvasásra a felhasznalonev, jelszo**.

## Fontos Függvények

```

1. void LoginLap(SDL_Renderer *renderer, gomb *gombok, alakzat *alakzatok, szo *szavak)

```

A LoginLap függvény **az eventek.c modulban** található, és a **main.c modul hívja** meg, akkor mikor a Login ablakot kell behozni, ez elindításkor, valamint zároláskor történik. A függvény további függvények segítségével **inicializálja a megjelenítendő grafikákat, valamint lefrissíti a képernyőt**. Paraméterek: \*renderer → ezt a main.c adja át cím szerint, ennek segítségével tudja az inicializálás végén **lefrissíteni a képernyőt**, \*gombok → címszerint veszi át a main.c ből a **gomb struktúra tömböt**, ezt tölti fel az **inicializált értékekkel**, \*alakzatok → az

**alakzatok tömböt** tölti fel inicializált értékekkel, \*szo → a **szo tömböt** tölti fel inicializált értékekkel.

```
1. void FelhasznaloLap(SDL_Renderer *renderer, gomb *gombok, alakzat *alakzatok, szo *szavak)
```

A FelhasznaloLap függvény az eventek.c modulban található, és a main.c modul hívja meg, tulajdonképpen **ugyanazokkal a paraméterekkel**, ugyan azt csinálja mint a LoginLap függvény, csak **más értékekkel inicializált** gombokat, alakzatokat, szavakat tesz a tömbökbe.

```
1. void ProgramEventek(SDL_Renderer *renderer, SDL_Window *window, gomb *gombok, int gombmeret, alakzat *alakzatok, int alakzatmeret, szo *szavak, int szomeret, bool loginaktivoldal, bool *programfut)
```

A ProgramEventek függvény sok paraméterrel rendelkezik, mivel, ez a főloopen **belüli loop**. Az **eventek.c modulban található**, és a **main.c modul hívja** meg. Ennek a loopnak a feladata, az egyes **felhasználói bemenetek** érzékelése, s ez alapján a helyes **függvény meghívása**, valamint az eventek utáni **képernyőfrissítés**. A program két esetben lép ki a loopból, egyrészt, ha **bezárjuk a programot**, másrészt ha **ablak váltás** történik. Ablak váltásnál az újonnan inicializált értékekkel **újrakezdi** a működést. Paraméterek: \*renderer → cím szerint veszi át a renderert, ez a pointer azért szükséges, hogy az egyes felhasználói eventek végén a függvény végre tudja hajtani a **képernyőfrissítést**, \*window → ez a pointer az **ablak műveletekhez** kell, minden esetben a méret a tömb méretét jelöli, \*gombok, gombmeret → a függvény legfőképpen ezekkel dolgozik, mivel, a gombokkal lehet érzékelni a **felhasználói eventeket**, valamint további **ablakspecifikus függvényekhez** kell, továbbá a **megjelenítő függvény** is megkapja, hogy ki tudja belőle olvasni a megjelenítéshez szükséges adatokat, \*alakzatok, alakzatmeret → **ablakspecifikus függvényekhez**, illetve a **megjelenítéshez kell**, \*szavak, szomeret → **ablakspecifikus függvényekhez**, illetve **megjelenítéshez kell**, loginaktivoldal → ez azt mondja meg a függvénynek, hogy éppen **melyik oldal az aktív**, melyik oldalnak a függvényei között kell keresni a helyeset, \*programfut → címként veszi át, mivel értéke lehet változni fog, **kilépés esetén** ez az érték hamisra állítódik, s megáll a loop.

```
void Kepernyofrissites(SDL_Renderer *renderer, gomb *gombok, int gombmeret, alakzat *alakzatok, int alakzatmeret, szo *szavak, int szomeret)
```

A megjelenites.c modulban található Kepernyofrissites függvényt az eventek.c modul hívja meg a felhasználói eventek után. Célja, hogy **összegyűjtse** az összes megjelenítendő gombot, alakzatot, szót, kiadja a **kirajzolási parancsokat**, majd **megjelenítse** az eredményt. Paraméterek: \*renderer → cím szerint veszi át, a **grafikus parancsokhoz** szükséges, minden esetben a méret a tömb méretét jelöli, a további pointerok és változók párban járnak, a függvény az odaadott méretek alapján **végigmegy a tömbökön**, s az engedélyezett gombokon, valamint az összes alakzaton, szón elvégzi a kirajzolási parancsokat.