Félkész Nagyházi Dokumentáció

**A Program Működése:**

A main modul egy “végtelen” loopból áll, ami addig fut, amíg nem adunk át “exit” commandot a programnak.

**A commandok kezelése** (a main egyetlen feladata az SDL ablak folyamatos frissítésén kívül):

A program a felhasználótól a standard inputra kap egy sztringet amelynek tartalmaznia kell a commandot és a kívánt paramétert space karakterrel elválasztva. Minden commandhoz egyetlen paraméter tartozik. A program ezután két külön sztringbe választja szét a kapott inputot, és amelyikbe a commandot tette, összeveti egy tömb elemeivel, amely a parancsok neveihez rendel egy függvénypointert. Ezután a kapott függvényt futtatja (mindegyik string paramétert kap, és saját maga alakítja át a kívánt tipusra).

**commands.c:**

*void GetCommand(char \*input):* Beolvas egy sort a stdin-ről és eltárolja egy stringben. Azért használok külön megírt függvényt erre, mert a commandok folyamatos beolvasásához fontos, hogy ne maradjon \n a stdin-on.

*void SeparateCommandLine(char \*input, char \*command, char \*argument)*: átadjuk ennek a függvénynek a *GetCommand* által kapott sztringet, és kettébontja. Az egyik a command, a másik a paraméter.

*void Open(char \*path):* A kép megnyitására használt command viselkedése. Ez a függvény két globális változót is beállít. Az imageInfo tárolja a kép felbontását, amelyet több modulból kell elérni. Az image a képet tárolja. A függvény először megállapítja a fájlról, hogy PPM formátumú-e az IsPPM függvény meghívásával. Amennyiben helyes a formátum, meghívja a LoadImageToArray függvényt, ami feltölti a képet egy tömbbe. Rendelkezik hibakezeléssel, pl nem lehet két képet egyszerre megnyitni, jelez, ha nem jó a fájl formátuma.

A többi függvény ebben a modulban a képfeldolgózó műveleteket hívja meg és alakítja át a paramétereket számukra. Mivel nem bonyultak, és majdnem ugyanúgy működik mind, nem szükséges őket részletezni.

**PPMhandling.c:**

*bool IsPPM(FILE \*imgfile):* A PPM fájlformátum szerint a fájl első két bájtját “magicnumber”-nek hívják. Ez határozza meg hogy PPM-e a fájl formátuma. A raw PPM P6-os magicnumbert használ, ez a függvény ezt ellenőrzi és ad vissza egy boolean értéket.

*Info PPM\_GetInfo(FILE \*imgfile):* Feltölti az imageInfo változót a megfelelő adatokkal. Szélesség, magasság pixelben, a maximális színérték. (a félkész verzió még csak 255ös maxcolorval-lal tud dolgozni)

*Pixel \*PPM\_LoadImageToArray(FILE \*imgfile, Info imgInfo, Pixel \*img):* Lefoglal egy tömböt a kép méreteivel, majd feltölti a fájlbol beolvasott adatokkal, és visszaadja a tömböt.

*Pixel \*PPM\_Darken(Pixel \*img, int amount, Info imgInfo):*Sötétíti a képet úgy, hogy minden pixel RGB értékeiből levonja a paraméterként kapott értéket. Ha minuszba menne egy színérték, helyette lenullázza azt.

*Pixel \*PPM\_Lighten(Pixel \*img, int amount, Info imgInfo):* világosítja a képet úgy, hogy minden pixel RGB értékeihez hozzáadja a paraméterként kapott értéket. Ha 255 fölé menne egy színérték, helyette 255re állítja azt.

*Pixel \*PPM\_Invert(Pixel \*img, Info imgInfo)*: negatív effekt. Minden RGB értéket kicserél a negáltjára (255-RGB).

*Pixel \*PPM\_Greyscale(Pixel \*img, Info imgInfo)*: Fekete-fehéríti a képet úgy, hogy átlagolja az RGB értékét minden pixelnek és minden színét erre az értékre állítja.

*Pixel \*PPM\_Contrast(Pixel \*img, Info imgInfo, int value):* Ez még nem működik. Placeholder. Nem is részletezem.

**system.c:**

Ebben a modulban található a pushmsg függvény ami csak annyit csinál hogy a kapott sztringet egy Paint.PPM: taggel ellátva kiírja a stdout-ra. Errorok és egyéb üzenetek közlésére használom, lehet átállok majd perrorra, megjátjuk. A többi függvény az SDL ablak elkészítésére és updatelésére szolgál, nem tartom fontosnak a részletezésüket, minden SDL-es programban megtalálhatóak ezek a kódrészletek, nem írtam ide semmi olyat ami kifelyezetten erre a programra lenne specifikus.

**Adatstrukúrák a programban:**

struct Pixel: előjeles karakter tipusokban tárolja el egy adott pixel RGB értékeit.  
struct Info: a megnyitott kép szélességét, magasságát és max színértékét tárolja int típusú változókban  
struct Command: ezt használom a kommandlista tömbhöz. Egy listaelem rendelkezik egy sztringben tárolt “id”-vel (ez alapján keresem meg) és egy func nevű függvénypointerrel ami a commandnak megegyező függvény címével rendelkezik.