

Kép beolvasására és kirajzolására alkalmas program megvalósítása.

Legyen egy Pixel struct, ami egy pixel értékét (r,g,b) tudja tárolni, valamint legyen egy Kepkezeló class, ami tartalmazza az aktuálisan beolvasott képet vector<vector<Pixel>> formában, valamint a kép méreteit két int típusú változóként.

A Kepkezeló tagfüggvényei legyenek a következők:

- Kepkezeló() konstruktor (valahogy inicializálja az int változókat, mindegy, hogy hogyan)
- Kepkezeló(string fajlnev) konstruktor, ami beolvassa az adott fájlt (a később definiált beolvas tagfüggvény segítségével).
- beolvas(string fajlnev) olvassa be a fajlnev nevű „.kep” fájlt a vector<vector<Pixel>>-be. Ehhez ki lehet használni, hogy ismerjük a kép méreteit (.kep fájl ezzel kezdődik, egyfajta headerként), de feltétlenül szükséges.
- kirajzol() kirajzolja a vector<vector<Pixel>>-t a képernyőre (itt derül ki igazából, hogy jó-e a beolvas() függvény).
- eltolva_kirajzol(int eltol_x, int eltol_y) eltolva kirajzolja a képet, a peremfeltétel pedig „toroid” (ami jobb oldalt kimegy, az baloldalt jön be, ami fent kimegy, az lent jön be ... a valóságban ehhez egy fánk alakú világ kéne, de programozásban bármi szabad ...).
- Lehetnek (de nem elengedhetetlenek) „getter” függvények, amik visszaadják a kép méreteit (jól jöhet a gin.open()-nél, amennyiben az a main()-ben van).

Amikor le van nyomva a bal egér gomb, a képet lehessen eltolni (azaz a kép legyen az aktuális egér pozíció mínusz a kiinduló egér pozíció nagyságával eltolva), ehhez kell az eltolva_kirajzol() függvény. Amikor nincs lenyomva a bal egérgomb, akkor a kép ugorjon vissza normál állapotba.

További funkciók megoldása esetlegesen újabb tagváltozók hozzáadásával és az eddigi tagfüggvények módosításával.

- Az „esc” billentyű lenyomásával zárjon be a képkezelő program
- Amíg le van nyomva a „space” billentyű, a kép negatívan jelenjen meg (és egyik eddig működő funkció se sérüljön)