Papp Benedek: GHQW4A

Feladat: Képfeldolgozás

Programozói dokumentáció

A program futtatása előtt gondoskodjunk a projektben használt külső könyvtárak meglétéről, valamint, hogy ezekből a megfelelő verziójú van telepítve. Hogy egyszerűbbé tegyem ezt a feladatot készítettem egy „requirements" nevű txt-állományt, ami tartalmazza az általam használt külső könyvtárak nevét valamint pontos verzióját. Ezek telepítését egyetlen paranccsal megteheti amennyiben windows operációs rendszert használ írja be a parancsorba: „python –m pip install –r requirements.txt", amennyiben linux operációs rendszert használ: „python3 –m pip install –r requirements.txt". Ezen kívül még használtam Czirkos Zoltán tanár úr konzolos megjelenítő modulját is amit nem talál meg a requirements-ben. (https://infopy.eet.bme.hu/pyconio/#1). Az oldalon pontosan leírja tanár úr miképpen kell a projektben elhelyezni a modult. (Ezek nem kerültek a feltöltött fájlok közé hely takarékosság miatt).

A main.py szerepe az alkalmazásban. A main gyakorlatilag csak egy belépési pontot jelöl meg a projektben. Nagyon hasonlít a menüben megírtakhoz ugyanis logikájában teljesen megegyezik csak a kód struktúrája miatt vettem külön a Main Screent. Ezen a fájlon belül van inicializálva az összes többi screen amik csak akkor jönnek létre ha a felhasználó kiválasztja az adott opció. Minden screen objektumnak van egy Start függvénye ami paraméterként kapja a jelenleg betöltött képet(amennyiben van ilyen) valamint az adott menüpontban felhasznált függvényeket amit az image fájl importálásával ér el.

A menu.py szerepe az alkalmazásban. Ebben a fájlban található szinte a teljes menü rendszer megvalósítsa(Kivéve a main-t). Először is kezdjük a osztályokkal 7db található benne minden almenünek van egy saját osztálya. Mindegyik osztálynak vannak alap adatai. Options => egy lista aminek az elemi az egyes menüpontok szövegesen. Azért választottam ilyesfajta tárolást, hogy elég legyen egy menü kiír függvényt csinálnom ami mindig paraméterként kapja a különböző menüpontokat. Ezene kívül még van egy MinChocie/MaxChoice attribútum is, amely arra hivatott hogy meghatározza azt az intervallumot amelyikből a felhasználó választhat. Mivel a konzolról való szám vagyis menüpont beolvasása nagyon gyakori feladat ezért erre is készítettem a függvényt, amely paraméterként kapja az intervallumot, beolvassa és ellenőrzi a beolvasott értéket amennyiben nem helyes hiba üzenetet ír ki a konzolra. Az olyan menü osztályok amelyeknek van további almenüje azt az osztály Start függvényében valósítottam meg. Függvények: PrintMenu(options, firstIgnored): Ez a függvény felel a menük kiírásáért, az első paraméter egy lista amely stringket tárol amelyek az egyes lehetőségek a menün belül, a másik paraméter arra szolgál ha szeretnék valamit közölni a felhasználóval pl. azt hogy mekkora jelenlegi kép stb. akkor ahhoz ne tartozzon szám. InvalidInput(): Igényesen kiírja a konzolra, hogy nem megfelelő az input. ImageNotLoaded(): Kiírja a konzolra, hogy még nincs betöltve kép. ValidateInput(MinChoice, MaxChoice): A megadott intervallumon belül fogad el számokat, amit beolvas int-ként adja vissza, ammenyiben nem felel meg a feltételeknek meghívja az InvalidInput() függvényt.

A image.py szere az alkalmazásban. Ebben a fájlban található a kép manipulálásához szükséges függvények, valamint egy osztály ami egy képet reprezentál. A kép osztály inicializálásánál megvizsgálom, hogy milyen operációs rendszert használ a felhasználó és annak függvényében mentem el a kép nevét, majd cv2 könyvtár segítségével beolvasom majd a kép mátrix reprezentációját elmentem a mtrx nevű attribútumba. Valamint a kép magasságát és szélességét is tárolom. Ebben az

állományban a függvényeket szerintem elég egyértelműen neveztem el (mit csinál a képpel) . Függvények: itt általánosan igaz az, hogy paraméterül kap egy img típusú objektumot és egy ilyennel tér vissza. Ez alól kivételt képez a kép megjelenítése mivel neki nincs vissza térési értéke. Az egyes függvényeket Log osztállyal dekoráltam. Ezt találtam a leg könnyebb megoldásnak arra, hogy minden manipuláció után rögzítve legyen a változás.

A log.py szerepe. Ebben a fájlban csak egyetlen osztály található amely egy dekorátor osztály. Csak annyit csinál, hogy amikor lefut egy olyan függvény amely ezzel az osztállyal van dekorálva, hozzá fűzi a logokat tartalmazó fájlhoz a függvény nevét amely meghívta. Így egyszerűen tudom követni az egyes képek változásait.