

Lazer Maze programozói dokumentáció

Készítette: Végh Iván PT9860

Projekt Felépítése és Szükséges Környezet

Fő Modulok:

main.py: Ez a fő futtatható fájl, amely a játék indításáért és a felhasználói interakciók kezeléséért felel.

logika.py: Ez a modul tartalmazza a játék logikáját és az adatmodelljeit, beleértve a profilokat, bábukat, és a játékmenet logikáját.

fajlkezeles.py: Ez a modul kezeli a fájlok olvasását és írását, például a pályák betöltését és a dicsőséglista kezelését.

Segéd Modulok:

pyconio: Egy külső könyvtár, amely konzolos alkalmazásokhoz nyújt különböző grafikus és interaktív funkciókat.

Adat Tárolás:

palyak/: Mappa, amely a játékpályákat tartalmazza (palya1.txt-től palya20.txt-ig).

profilok/: Mappa, ahol a játékosok profiladatai vannak tárolva (gamertag.txt formátumban).

dicsoseglista.txt: Fájl, amely a játékosok dicsőséglistáját tartalmazza.

Szükséges Környezet és Telepítés:

Python: A projekt Python 3.x verzióban íródott, ezért szükséges a Python legújabb stabil verziójának telepítése.

Külső Könyvtárak: A pyconio könyvtár telepítése szükséges, amelyet a Python csomagkezelőjével (pip) telepíthetünk a következő paranccsal: `pip install pyconio`.

Telepítési Lépések:

Nyisd meg a terminált vagy parancssort a projekt mappájában.

Telepítsd a pyconio könyvtárat a `pip install pyconio` parancs kiadásával.

Futtasd a `main.py` fájlt a Python interpreterrel.

Megjegyzés: A projekt nem igényel további külső könyvtárakat vagy környezeti beállításokat. Az összes szükséges adatot a fenti mappaszerkezetben és fájlokban tárolja.

Adatszerkezetek Dokumentációja és Tervezési Megfontolások

Profil Osztály:

Célja: A játékosok profiladatainak tárolása, mint például a gamertag, a teljesített pályák, az összpontszám és az összesen eltöltött idő.

Tárolt Adatok:

gamertag (str): A játékos egyedi azonosítója.

teljesített_palyak (lista): A teljesített pályák listája, TeljesítettPalya objektumokat tartalmazva.

osszpontszam (int): A játékos által összegyűjtött pontok száma.

osszido (int): A játékban eltöltött idő másodpercben.

Tervezési Megfontolások: A profil osztály lehetővé teszi a játékosok adatainak strukturált és egységes tárolását, megkönnyítve ezzel az adatok kezelését és a fájlba írást/olvasást.

Babu Osztályok:

Célja: A játéktáblán található különböző típusú bábuk reprezentálása és kezelése.

Alapvető Típusok:

Ures: Az üres mezőket jelöli.

SpaceRock: Akadályokat reprezentál, melyek blokkolják a lézer útját.

Tukor: A lézer irányát megváltoztató elem.

AteresztóTukor: Speciális tükör, amely átereszti és tükrözi is a lézert.

Raketa: Célbábuk, amelyeket a lézerrel kell aktiválni.

Tervezési Megfontolások: Minden bábu típusnak saját osztálya van, amely lehetővé teszi a különböző viselkedések és tulajdonságok definiálását. Ez a megközelítés rugalmasságot és könnyebb bővíthetőséget biztosít.

Tabla Osztály:

Célja: A játéktábla reprezentálása, amely a bábukat tartalmazza.

Tárolt Adatok:

matrix (lista listája): A táblát reprezentáló 5x5-ös mátrix, Babu objektumokat tartalmazva.

Tervezési Megfontolások: A mátrix alapú tárolás lehetővé teszi a tábla egyszerű ábrázolását és a bábuk pozícióinak könnyű kezelését.

TeljesítettPalya Osztály:

Célja: Egy teljesített pálya adatainak tárolása.

Tárolt Adatok:

id (int): A pálya azonosítója.

nehezseg (str): A pálya nehézségi szintje.

ido (int): A pálya teljesítéséhez szükséges idő.

Tervezési Megfontolások: Az osztály segítségével strukturáltan tárolhatóak a teljesített pályák adatai, ami megkönnyíti a profilokban történő kezelést és a statisztikák generálását.

Adat Tárolási Formátumok:

Célja: A játékhoz szükséges adatok tárolása fájlokban.

Típusok:

.txt fájlok a pályák, profilok és a dicsőséglista számára.

Tervezési Megfontolások: A szöveges formátum könnyű olvashatóságot és szerkeszthetőséget biztosít, ami megkönnyíti a pályák és profilok kezelését, valamint a dicsőséglista frissítését.

Függvények/Metódusok

main.py függvényei

`print_fejlec(profil: logika.Profil = None, palya_id: int = None, palya_nehezseg: str = None) -> None:`

Feladat: Megjeleníti a játék fejlécét, beleértve a verziószámot, a játékos gamertagját és a pályainformációkat, ha azok rendelkezésre állnak.

Paraméterek:

profil (logika.Profil): A játékos profilja. Alapértelmezésben None.

palya_id (int): A pálya azonosítója. Alapértelmezésben None.

palya_nehezseg (str): A pálya nehézségi szintje. Alapértelmezésben None.

Visszatérés: Nincs visszatérési érték.

`print_tabla_lezzerrel(tabla: logika.Tabla, laser_utvonalak)-> None:`

Feladat: Megjeleníti a játéktáblát a képernyőn, beleértve a bábukat és a lézer útvonalait.

Paraméterek:

tabla (logika.Tabla): A játéktábla objektuma.

laser_utvonalak: A lézer útvonalainak listája vagy egyéb reprezentációja.

Visszatérés: Nincs visszatérési érték.

`print_dicsoseglista()-> None:`

Feladat: A dicsőséglistát jeleníti meg, amely rangsorolja a játékosokat a teljesítményük alapján.

Paraméterek: Nincsenek.

Visszatérés: Nincs visszatérési érték.

`print_raketak_info(raketak: list)-> None:`

Feladat: Megjeleníti az összes rakéta információit a játék során.

Paraméterek:

raketak (lista): A játékban lévő rakéták listája.

Visszatérés: Nincs visszatérési érték.

`print_felhasznalhato_babuk(felhasznalhato_babuk: list)-> None:`

Feladat: A felhasználható bábuk listáját jeleníti meg.

Paraméterek:

felhasznalhato_babuk (lista): A játékban használható bábuk listája.

Visszatérés: Nincs visszatérési érték.

`jatek_futtatas(profil: logika.Profil, palya_fajlnev: str)-> None:`

Feladat: Egy játékmenetet futtat megadott profil és pályafájl alapján.

Paraméterek:

profil (logika.Profil): A játékos profilja.

palya_fajlnev (str): A pályafájl neve.

Visszatérés: Nincs visszatérési érték.

`profil_menu(profil: logika.Profil)-> None:`

Feladat: Egy menüt jelenít meg az adott profilhoz, ahol a játékos választhat a különböző opciók között.

Paraméterek:

`profil (logika.Profil):` A játékos profilja.

Visszatérés: Nincs visszatérési érték.

`main()-> int:`

Feladat: A főmenüt és a program fő végrehajtási ciklusát tartalmazza.

Paraméterek: Nincsenek.

Visszatérés: 0, ami a program sikeres befejezését jelzi.

logika.py Függvények és Osztálymetódusok

`irany_inverter(lezer_irany: int)-> int:`

Feladat: Megfordítja a megadott lézer irányát.

Paraméterek: `lezer_irany (int):` A lézer jelenlegi iránya.

Visszatérés: Az irány megfordítva (int).

Profil Osztály:

`frissit_palya_eredmenyt(self, palya_id: int, palya_nehezseg: str, uj_ido: int)-> bool:`

Feladat: Frissíti a megadott pálya eredményét a profilban.

Paraméterek: `palya_id (int), palya_nehezseg (str), uj_ido (int).`

Visszatérés: True, ha frissítés történt, egyébként False.

Tukor Osztály:

`tukrozes(self, beerkezo_irany):`

Feladat: Kiszámítja a lézer új irányát a tükörre érkezéskor.

Paraméterek: `beerkezo_irany:` A lézer érkezési iránya.

AteresztoTukor Osztály:

tukrozes_ateresztes(self, beerkezo_irany):

Feladat: Kiszámítja a lézer új és áteresztett irányát.

Paraméterek: beerkezo_irany: A lézer érkezési iránya.

Tabla Osztály:

babu_elhelyez(self, babu: "Babu", pozicio: tuple)-> None:

Feladat: Elhelyez egy bábút a táblán.

Paraméterek: babu (Babu), pozicio (tuple).

lezer_kovetkezo_szamitas(pozicio: tuple, irany: int)-> tuple:

Feladat: Kiszámítja a következő pozíciót a táblán a megadott lézer irány alapján.

Paraméterek: pozicio (tuple), irany (int).

Visszatérés: Az új pozíció (tuple).

lezer_utvonal(tabla: "Tabla", kezdo_pozicio = (1, 0), kezdo_irany = 1)-> dict:

Feladat: Kiszámítja a lézer útvonalát a táblán.

Paraméterek: tabla (Tabla), kezdo_pozicio (tuple), kezdo_irany (int).

Visszatérés: Az útvonalak (dict).

lezer_irany_kiszamitas(jelenlegi_pozicio: tuple, eloza_pozicio: tuple)-> int or None:

Feladat: Kiszámítja a lézer irányát két adott pozíció alapján.

Paraméterek: jelenlegi_pozicio (tuple), eloza_pozicio (tuple).

Visszatérés: A lézer iránya (int) vagy None.

get_raketak(tabla: Tabla)-> list:

Feladat: Visszaadja az összes rakétát a tábláról.

Paraméterek: tabla (Tabla).

Visszatérés: A rakéták listája (list).

`raketa_aktivalas_deaktivallas(tabla, lezer_utvonalak)-> Tabla:`

Feladat: Frissíti az összes rakéta aktiválási állapotát a táblán.

Paraméterek: `tabla` (Tabla), `lezer_utvonalak` (dict).

Visszatérés: A frissített tábla objektum (Tabla).

`minden_babu_felhasznalva(tabla: Tabla, felhasznalhato_babuk: list)-> bool:`

Feladat: Ellenőrzi, hogy minden felhasználható bábu el lett-e helyezve.

Paraméterek: `tabla` (Tabla), `felhasznalhato_babuk` (list).

Visszatérés: True, ha minden bábu el van helyezve, különben False.

`interakcio_a_babukkal(tabla: Tabla, felhasznalhato_babuk: list)-> Tabla:`

Feladat: Kezeli a játékos interakcióját a bábukkal.

Paraméterek: `tabla` (Tabla), `felhasznalhato_babuk` (list).

Visszatérés: A frissített tábla objektum (Tabla).

`kovetkezo_palya(profil: Profil)-> None or str:`

Feladat: Kiválasztja a következő játszandó pályát a profil alapján.

Paraméterek: `profil` (Profil).

Visszatérés: A következő pálya elérési útvonala vagy None.

`korabbi_palyak_kivalasztasa(profil: Profil)-> None or str:`

Feladat: Lehetővé teszi a korábbi pályák kiválasztását.

Paraméterek: `profil` (Profil).

Visszatérés: A kiválasztott pálya elérési útvonala vagy None.

`fajlkezeles.py` függvényei

`beolvas_palyat(fajlnev: str)-> tuple:`

Feladat: Beolvassa és feldolgozza a megadott fájlban található pálya adatait.

Paraméterek: `fajlnev` (str): A beolvasandó pálya fájljának neve.

Visszatérés: Egy tuple, amely tartalmazza a létrehozott táblát, a felhasználható bábuk listáját, valamint a pálya azonosítóját és nehézségi szintjét.

`dicsoseglista_keszitese_frissites()` -> None:

Feladat: Frissíti a dicsőséglistát a profilok mappában található összes profil adatai alapján.

Paraméterek: Nincsenek.

Visszatérés: Nincs visszatérési érték.