

jelmagyarázat:

Félkövér, sárgával írott a programkódrészlet, **Fekete háttérrel a kiemelt részlet;**

Félkövér, kékkel a modulnevek;

Aláhúzott, vörössel a függvénynevek;

lilával azon magyarázatok, melyek nem közvetlenül az adott téma, cím alá tartoznak, ám azok magyarázatához szükséges a tartalmuk.

zölddel a változónevek

A program menüvezérelt, 4 modullal kiegészítve:

-a program a pygame multimédiás modult használja.

A menüvezérlésért két ciklus felelős: A **főprogram** főciklusa, mely megszakítását a programban több menüből elérhető 'KILÉPÉS' gomb megnyomása, vagy az ablak bezárása eredményez. Ezen ciklus felelős a különböző menükben, a „**menu_kezelo**” modul által meghatározott utasítások végrehajtásáért, illetve azok végrehajtásához szükséges függvények meghívásáért. A második ciklus „**menu_kezelo**” modul „megnyit” függvényének ciklusa. Ez a ciklus a játékos kattintásait vizsgálja. „nev” azaz név paramétere egy menü neve, ezen menünev alapján vizsgálja, hogy a játékos milyen pozícióra kattintott, és hogy ezen pozíció állapotváltozást kell, hogy hozzon. Ha történik ilyen kattintás, akkor a program egy sztring, int, vagy tuple értékkel tér vissza. Ezen visszatérési értékek utasítások a főciklus felé. Példa:

```
if nev == 'fomenu':  
    if benne(x, y, 100, 700, 100, 270):  
        kirajzol('palyamenu', window)  
        return 'palyamenu'  
    elif benne(x, y, 100, 700, 430, 600):  
        return 'KILEPES'
```

A kódrészlet a megnyit függvény ciklusának azon ága, mely a játék főmenüjének megnyitását követően fut le. ha a függvény ezen ágba tartózkodik, csak akkor fut le, ha a játékos kattintásának x és y pozíciójára igaz az $x_{min} < x < x_{max}$ és $y_{min} < y < y_{max}$ állítás. Grafikusan ez két gomb valamelyikébe való kattintás jelent. Ha a játékos a kilépés gombra kattint, a feketével kiemelt ágba lép a program, és 'KILÉPÉS' értékkel tér vissza, mely a főciklusban ciklustörést eredményez.

A megnyit függvény változatosan viselkedik.: Egyes menükben, mint: 'fomenu', 'jatekmenu', 'gyozelem', 'vereseg', a kattintható pozíciók egyesével vannak meghatározva, előre megadott visszatérési értékkel, amely vagy egy újabb megnyitandó menü neve (ezek kisbetűvel írottak) például 'fomenu', 'jatek-kepernyo', vagy egy utasítás (nagybetűs), mint a 'KILEPES'. A 'palyamenu' név paraméter esetén egyrészt azt vizsgálja a program, hogy egymás alatt 100-100 pixel távolságra lévő gombok közül melyikre kattintott a felhasználó, ezen integer típusú (1, 2, 3, 4, 5) értékeket felvehető érték egy sztringhez lesz hozzáfűzve, mely ezen menüben kiválasztott pálya azonosítására szolgál. A 'jatek_kepernyo' esetén nem csak gombok kerülnek kirajzolásra, hanem itt történik maga a játék is, így a ciklus ezen ága visszatérhet egy sztringgel, egy gomb megnyomásakor, vagy integerrel, amely egy eszköztári elemre való kattintáskor az elem sorszámát adja vissza, illetve visszatérhet tuple-vel, mely a játéktér, vagy pálya egy adott cellájának x és y koordinátáját tartalmazza.

Modulok:

- **BLOKK:** ez a modul csupán egy osztályt tárol (Blok), paraméterei az irány, és fajta. Ilyen típusú objektumok a játék cellái, fajtájuk, és elforgatási irányuk alapján kerülnek azonosításra. két speciális metódusa az egyenlőség és különbözőség. Két Blokk akkor egyenlő, ha egyezik fajtájuk, és elforgatási irányuk párossága. Ezen összehasonlításokat csupán olyan celláknál használ a program, melyeknél az elforgatás irányának csupán párossága számít. Két blokk akkor nem egyenlő, ha az egyenlőség nem teljesül.
- **menu_kezelo:** a modul **'benne'** függvénye egy két számról határozza meg, hogy két-két szám közt helyezkednek-e el, visszatérési értéke logikai, megadja hogy egy adott pont belül van-e, egy két szemközti csúcsponttal megadott téglalaton. **'megnyit'** függvénye a már tárgyalt módon működik, **'kirajzol'** függvénye pedig a 'nev' paraméterként megadott nevű fájlt tölti be a programkönyvtárból, majd jeleníti azt meg, a 'window' paraméterben megadott felületen, amely a játék ablaka. Ilyen módon van megjelenítve minden menü.
- **palya_valaszt:** a modul egy függvénnyel rendelkezik, és használja a BLOKK modul Blokk osztályát. **'betolt'** függvényét a **főprogram** főciklusa hívja meg, ha az **aktual_menu** változó értéke 'palya1', vagy 'palya2' ... 'palya5', mely értékek a **menu_kezelo** modul **'megnyit'** függvényének 'nev'='palyamenu' paraméterezésének esetén lehetséges visszatérési értékei. Ezen értékek jelölik a játékban betölthető 5 különböző pályát. Ezen pályák pályanév.txt formában mentettek (pl: 'palya3.txt'), fájlként két sort tartalmaznak.: az első sor, a játéktér kezdeti blokkjait tárolja, az alábbi módon: egy blokkot egy, vagy két karakter jelöl: 'u' = üres cella, 'v' = célpont, 'b' = blokk/akadály, és 's0', 's1', 's2', 's3', amelyek a start cellát, rendre 0, 1, 2, vagy 3 elforgatási értékkel. A játéktér 5*5 cellából áll, a fájlban 5 darab, darabonként 5 cellát jelölő karakterlánc van szóközzel elválasztva. Minden ilyen 5-ös egység egy sort jelent a játéktérben. A második sor tartalmazza az eszköztár elemeit.: ezek értéke 't' = tükör és 'f' azaz féligáteresztő-tükör. Ezekből legkevesebb 1, és legfeljebb 6 darab lehet összesen. Ezen elemek egymástól el-nem választva vannak lejegyezve. A játéktér listában lesznek tárolva a játéktér sorai, az eszköztár listában pedig az eszköztári elemek. A függvény kiegészíti a nevet '.txt'-kiterjesztéssel, majd megnyitja a fájlt. A fájlban lévő karakterek segítségével létrejön egy 5 elemű, egyenként 5 elemű, Blokk típusú objektumokat tartalmazó lista, mint játéktér, majd egy max 6 elemű, Blokk típusokat tartalmazó eszköztár. Visszatérési értékei ezek, illetve a játéktér y és x indexe, mely a start cellát jelöli, tuple-ben.
- **jatek_kezelo:** függvényenként:
 - o **fajtabol_kep:** paraméterként egy Blokk fajtanevét kapja, majd visszatér a Blokknak megfelelő képpel. A képek azonos nevek a fajta paraméter felvehető értékeivel.
 - o **eredmeny:** egy duplaciklussal megvizsgálja a játéktér összes celláját, és ha valamely cellában tárolt Blokk 'fajta' paramétere 'stop' értékű, False, egyéb esetben True értékkel tér vissza. Ezen érték a játék győzelmének értéke. Működése később nyer értelmet, de egyelőre azért helyes, mert ha egy 'stop' azaz célpont blokkot eltalál egy lézer, annak fajta értéke 'be_stop'-ra változik, azaz egy jól teljesített pálya esetén nincs lézerrel el nem talált célpont cella.
 - o **blokk_kirajzol:** egy blokkot rajzol ki a képernyőre, a Blokk.fajta paraméterrel meghívja a **fajtabol_kep** függvényt, majd a kapott képet, Blokk.irany paraméter-szer 90 fokkal elforgatva (órmutató járásával ellentétes irányba) beilleszti a képernyő x és y koordinátájára, mely x és y értékeket paraméterként kapja.

- **palya kirajzol:** a játéktér összes elemére meghívja a **blokk kirajzol** függvényt, mely y koordinátája a sorszám azaz külső tömb indexe, x koordinátája pedig az oszlopszám, azaz belsőbb listák indexei, megfelelő eltolással.
- **eszkoztar kirajzol:** az eszköztár elemein végighaladva meghívja a **blokk kirajzol** függvényt, az eszköztár elemekre, melyekkel alulról tölti fel az eszköztár területét.
- **eszkoztar elem kijelöl:** Az eszköztár egy elemének sorszámát kapva paraméterül, megjelenít egy zöld keretet az adott elem körül. Ez a jelzi vissza, hogy egy cella ki lett jelölve az eszköztárban.
- **palya blokk kijelöl:** A játéktér egy cellájának sor és oszlopszámát kapva paraméterül kijelöli az ezen paramétereknek megfelelő cellát, szintén zöld keretet kap.
- **tovabbit:** paraméterként kapja a játéktér, egy x és egy y koordinátát, azaz sorszámot és oszlopszámot, illetve egy irányt. A játéktér[y][x] indexű Blokkjának fajtájának és irányának, illetve az irány paramétereknek megfelelően több módosítást hajthat végre a játéktér, majd visszatér játéktér értékével.
Módosítások: Lecserélheti játéktér[y][x] cellában lévő Blokkot, meghívhatja magát egy másik y, x, és irány paraméterrel. Értelmezése: a jatekter[y][x] celláját lézer érte el irány irányból, kérdés, hogy ez hatással van-e a Blokk fajtájára, és elforgatási irányára, illetve, hogy továbbítani kell-e a lézert, és ha igen, milyen irányba? Az irány értéke 0,1,2,3 lehet, mely azt jelenti, hogy a lézer rendre: alulról, jobbról, felülről, vagy balról érkezett. az irányok listában olyan x és y párok vannak tuple-be megadva, mely listának irány-val indexelt eleme az iránynak megfelelő következő koordinátára mutat. Ha x, vagy y értéke nem 0 és 4 közti, akkor a lézer elhagyta a pályát, ilyenkor nem történik módosítás, a függvény visszatér jatekter értékével. Ha ez nem áll fenn, a Blokk fajtájától függően a Blokk.irany, azaz elforgatas és irány felhasználásával eldől, hogy lesz-e cserélve a Blokk fajtája és iránya, majd, hogy merre továbbítja a lézert a cella. Féligáteresztő tükrök esetén két különböző fajtájú Blokkra cserélődhet a Blokk, ennek oka, hogy míg a féligáteresztő tükrök képe átlósan szimmetrikus, azaz 2 különböző megjelenése lehet, addig mindkét megjelenése a 4 lehetséges irányú lézer beérkezésekor 4-4 különböző képet kell, hogy felvegyen. Ezért van `I_laser_f_tukor` és `I_laser_f_tukor_inverz` fajta is.
- **főprogram/Laser-maze:** a grafikus felület beállítása után kirajzoltatja a főmenüt, majd az aktual_menu értékadására meghívja a menu_kezelo.megnyit(...), függvényét, amely addig várakozik, míg a játékos nem kattint egy gombra. Ha a játékos kattint egy érvényes pozícióra, az aktual_menu értéke egy utasítás lesz. Ezen utasítás minden ciklusban végrehajtódik, és új értéket kap, mely szintén a menu_kezelo:megnyit(...) megfelelően paraméterezett függvény visszatérési értéke lesz. Ha ez az utasítás egy az alábbiak közül: ['fomenu', 'palyamenu', 'jatekmenu', 'gyozelem', 'vereseg'], ki kell rajzolni az utasításnak megfelelő menüt, majd várakozni a játékos kattintására az új menüben. Más esetben, ha a 'palya' sztring benne van az utasításban, akkor egy pálya nevét kaptuk vissza, tehát létre kell hozni a játékhoz szükséges tárolókat (jatekter, eszkoztar, start), létre kell hozni egy változót arra az esetre, ha a játékos ki akar választani egy elemet a pályán, ki kell rajzolni mindent, majd várakozni a játékos kattintására a jatek_kepernyo menüben. SZIMULACIO esetén a játéktér le-kell szimulálni a lézer közlekedési irányát, ezt a jatek_kezelo.tovabbit függvényének meghívásával lehet, melyet start cellából indítunk, annak elforgatásával megegyező értékű irány-ba. A pálya kirajzolása után, a jatek_kezelo.eredmeny értékétől függően a győzelem, vagy vereség menü jelenik meg. a FOLYTAT utasítás a jatek_menu -ból a jatek_kepernyore visz vissza. Ha az utasítás int típusú, akkor a játékos egy elemet választott az eszköztárból: Ha ez az elem nem az

eszköztár egy üres helyére mutat, meg kell vizsgálni, hogy előzetesen választottunk-e elemet.: A választott változó értéke eredetileg None, ha értéke None, és aktual_menu egy intet adott vissza, akkor a választott egy tuple lesz, mely első értéke az eszköztár választott elemének indexe, második pedig 'eszközt.' sztring, mely azonosítja, a választás helyét. aktual_menu ha tuple értékű, akkor a jatekter-rol történt a választás. Ha a választott értéke nem None, és a választott elem a játékszabályoknak megfelelően cserélhető, esetleg lerakható aktual_menu által mutatott elemre/elemmel, akkor a csere megtörténik, az érintett elemek újra kirajzolásra kerülnek. Ha választott értéket kapott, utána mindig meghívódik a választás kategóriájának megfelelő kijelöl függvény is, (jatek_kezelo.eszkoztar_elem_kijelol(...); jatek_kezelo.jatekter_blokk_kijelol(...)). FORGAT utasítás esetén ha választott-nak nincs értéke, nem történik semmi, egyébként, ha játéktéren van választott elem, akkor az lecserélődik egy olyan Blokkra, mint önmaga volt, viszont elforgatása egyel nő. KILEPES utasítás esetén a ciklusfeltétel nem teljesül, a ciklus megszakad, majd meghívódik a pygame modul quit() függvénye, a program kilép.

-

- A főprogram (Laser .py)-ben történik a pygame inicializálása, a játék ablakának létrehozása. Ezt követi a menu_kezelo modul kirajzol függvényének meghívása 'fomenu' és window paraméterekkel, melynek hatására megjelenik a főmenü. Ezek után az aktual_menu változó értéke 'fomenu' lesz. Ezen változó tárolja a menü

Menüvezérlés