

# Farmos játék programozói dokumentáció

## Modulok

A program a fő funkciók alapján lett a következő főmodulokra felbontva: main, game, scoreboard és credit. Mindegyik modul saját feladatot lát el a programom belül, különböző segédmodulok és könyvtárak segítségével.

## Pygame

A program a Pygame könyvtárat használja. - <https://www.pygame.org/>

## Main

Ez a modul a főprogram, a legfőbb feladata a menürendszer összerakása. Használja a button.py modult, a gombok megjelenítéséhez.

## Működés

A main.py nevű fájlban hívódik meg a main függvény, ami tartalmazza az egész programot. Ebben a függvényben deklarálódik a program 2 legfontosabb eleme: a kijelző és az óra

A kijelző a program kimeneti felülete

Mivel a program folyamatos futásához végtelen while ciklusra van szükség, amitől instabillá válhat a rendszer, ezért alkalmazok egy órát, aminek tick nevű tagfüggvényének hívásával (clock.tick(60)) szinkronná teszem a ciklust.

A main.py modul tartalmaz egy set() függvényt, ami megváltoztatja egy gomb felszínét, kinézetét. Paraméterként bekéri azt a sprite csoportot, ami tartalmazza a gomb egyik állapotSprite-ját, egy MenuButton típusú változót, illetve az új állapot kódját: 0 vagy 1. Ez a függvény akkor van hívva, ha a felhasználó ráviszi az egerét egy gombra. Ebben az esetben a MenuButton és a Spritegroup tagfüggvényei felhasználásával megváltoztatja a gomb kinézetét. A megjelenítő csoportból eltávolítja a régi állapotot, és visszateszi helyére az újat. A gomb így grafikusan megváltozik a kijelzőn

A main.py-ból hívódik meg a többi főmodul main függvénye. Egy ilyen hívás esetén ezek a függvények paraméterként megkapják a két legfontosabb elemet: a kijelzőt és az órát.

## Game

A játék modul és a main.py főmodul között helyezkedne el a newGame.py modul, ami a tesztelések miatt nem szerepel a programban.

## Függvények:

### Main():

A main függvény paraméterként megkapja a kijelzőt, és az órát a játék befejeztével visszatér a főprogramba.

Ebben a modulban is a játékot egy végtelen ciklus hajtja, ami az órával van időzítve. A cikluson belül adatbekérések és feltételek futnak le amik alapján a függvények illetve a tagfüggvények meghívódnak.

### Mezokeres():

Paraméterként bekér egy 2 értékű tuple-t vagy listát, ami egy síkbeli pozícióval egyezik meg, illetve egy kert objektumot, amiben keresni fog parcellákat.

A kert objektum `cellakeres()` tagfüggvényét segítségül hívva visszatér egy táblázati értékkel, ami a parcella/mező kerten belüli koordinátája.

*Buy():*

Ez a függvény egy kiválasztott növény típust helyez le egy parcellára a parcella `ultet()` tagfüggvényével.

Paraméterként bekéri egy parcella kertbeli koordinátáit és a kert objektum `get()` tagfüggvényével megkeresi a parcellát. Bekér továbbá egy növénytípust (osztálynevet) illetve egy kert objektumot.

A függvény nem tér vissza semmilyen értékkel. A függvény lehelyezi a növényt a parcellához.

*idolgeny():*

Paraméterként bekér egy növény típusú objektumot.

Visszatér a következő növésig szükséges időtartammal.

*vizlgeny():*

Paraméterként bekér egy növény típusú objektumot.

Visszatér egy bool értékkel, ami meghatározza, hogy a növényt a jelenlegi állapotában kell-e locsolni.

*novenyNoves():*

Paraméterként bekér egy kert objektumot, illetve egy Spritegroup-ot, amiben a növés véghez fog menni.

Nincs visszatérési értéke. A függvény végigmegy az összes növényen, amit a megadott csoport tartalmaz, és az `idolgeny()` illetve a `vizlgeny()` függvények segítségével megnöveszti/elrohasztja azokat, vagy felszólítja a játékost, hogy öntözze meg az adott növényt.

*Vizcsepp():*

Paraméterként bekér egy növény objektumot, illetve egy megejelenítő szintet (Spritegroup).

Nem tér vissza értékkel. Megjelentet egy vízcseppet az összes locsolandó növény bal felső sarkában.