

MAYA

A játék leírása (fejlesztői dokumentáció)

A main() függvény, és globális paraméterek:

Globális paraméterek:

- *int reg*: két értéket vehet fel: 0: A az aktuális játékos, 1: B az aktuális játékos
- *int volt*: két értéket vehet fel: 0: pakolhatunk a godorbe, 1: nem pakolhatunk a godorbe
- *char ch*: univerzális karakter input beolvasására
- *int olv*: az aktuálisan kijelölt doboz száma
- *int p*: annak a doboznak a száma, amelyikbe a lépkedés során éppen pakolunk
- *char szam[2]*: a golyók számát char-rá alakítva ebben a változótömbben tároljuk
- *int vege*: ha 1, akkor vége a játéknak

A main() függvény:

- betölti a grafikus eszközöket
- kirajzol egy rövid ismertetőt a játékról
- gombnyomásra elkezdi a játékot:
- létrehoz egy példányt a *Tabla* osztályból, és kirajzolja azt
- egymás után hívja meg a *Rajzol()*, *beolvasas()*, *lepes()* és *ellenorzes()* függvényeket, amíg a *vege* változó 1 nem lesz
- ha vége, akkor megkérdezi, hogy lesz-e új játék, és ha igen, akkor előlről kezd (az ismertető kihagyásával)

Objektum-orientáltság, avagy az osztályok felsorolása, és kapcsolatuk ismertetése:

- *Godor* osztály: Ez az osztály valósítja meg a játékosok gyűjtőgödreit. Két példány jön létre belőle, melyek a következő paraméterekkel rendelkeznek:
 - *golyok*: a godor által aktuálisan birtokolt golyók száma
 - *regi*: a godor előző állapotában a golyók száma
 - *color*: a godor színe
 - *px*: a godor balfelső sarkának x koordinátája
 - *py*: a godor balfelső sarkának y koordinátája
 - Függvényei a következők:
 - *Godor()*: konstruktor - 0 golyóval és adott paraméterekkel létrehozza a godrot
 - *GetGolyok()*: Visszaadja a godor golyóinak számát
 - *Berak()*: Egyel megnöveli a golyók számát
 - *Kirajzol()*: Kirajzol a megfelelő helyre egy adott színű godrot, és bele rajzolja a golyók számát
- *Doboz* osztály: Ez az osztály valósítja meg a játékosok dobozait. 6*2 példány jön létre belőle (6 kék és 6 piros). Örökli a *godor* osztály paramétereit és függvényeit egy-két kiegészítéssel:
 - 6 golyóval rendelkező kezdetben
 - *Kivesz()*: a golyók számát eggyel csökkenti

- *InverzKirajzol()*: Kirajzol a megfelelő helyre egy zöld színű dobozt, és bele rajzolja a golyók számát (ez az éppen kijelölt doboz lesz)
- *Tabla* osztály: Tulajdonképpen egy az osztály valósítja meg az egész játékot. Mutatókkal rendelkezik két tömbre, melyek a dobozokat és godroket tartalmazzák. Konstruktorában létrehozza a játékhoz szükséges objektumokat, *Rajzol()* metódusával pedig kirajzolja azt.

A *Tabla* osztály metódusai, avagy a programot működtető algoritmusok ismertetése:

- *hirdet(int)*: annak megfelelően, hogy 0 (kék), 1 (piros), vagy 2 (döntetlen) értéket kap paraméterként, kiírja a képernyőre az eredményt (fehér keretben).
- *beolvasas()*: a beolvasott karakternek megfelelően változtatja az olv értékét (reg ismeretében), és mindig kirajzolja az aktuálisat zölddel, a régit pedig a megfelelő színnel. Mindezt az E billentyű lenyomásáig folytatja.
- A *lepes()* függvény foglalja magában a szabályokat, vagyis ezek ismeretében hajtja végre a golyók rendezését:
 - *int g*: itt tároljuk a kezdeti golyók számát
 - *int temp*: itt tároljuk a kezünkben levő golyók számát
 - *int p*: itt tároljuk a megjelölt doboz számát.
 - Csak akkor folytatjuk, ha ez nem 0
 - *p* értékét megnöveljük (mod 12)
 - Ha pont a godor előtti doboznál járunk ($p=6$ és $reg=0$, vagy $p=0$ és $reg=1$) és pakolhatunk a gödörbe ($volt=0$), akkor kiveszünk egy golyót a kezünkből (*temp*), hozzáadunk egyet a gödörhöz, és ha már nincs több golyó benne, az azt jelenti, hogy megint ugyan az a játékos következik (és megint pakolhatunk a godorbe) ha még van golyó a kezünkben, akkor most már nem pakolhatunk a godorbe, hanem az ellenfél doboza következik. Ezután csökkentjük a *p*-t visszaállítva azt a godor előtti utolsó dobozra.
 - Ha nem ilyen helyzet alakul ki, akkor a *volt*-ot visszaállítjuk 0-ra (hogy ha majd megint az utolsó dobozhoz érünk, akkor már pakolhassunk bele), az aktuális gödörbe bepakolunk, kezünkből pedig kiveszünk egy golyót.
 - Kiveszünk az eredeti dobozból pontosan annyi golyót, amennyi eredetileg benne volt így pont annyi marad benne, amennyit útközben beleraktunk
 - Ezután jöhet a másik játékos (tehát a „kétszer jövést” úgy oldottam meg, hogy kétszer is megváltoztattam az aktuális játékost)
 - Ha kész, akkor az *olv*-ot ráállítjuk az aktuális játékos első dobozára
- *ellenorzes()*: 0-ról indulva összeszámolja a játékosok dobozaiban levő golyókat, és ha az egyik 0, az azt jelenti, hogy már nem tud lépni, ekkor a játék véget ér. Meghívódik a *hirdet()* függvény annak megfelelő paraméterrel, hogy kinek a godreben van több golyó. Ha sor kerül eredményhirdetésre, akkor visszatér 1-el, így a vege változót is beállítva, ha nem, akkor 0-val tér vissza.