Arcade games- Snake Game

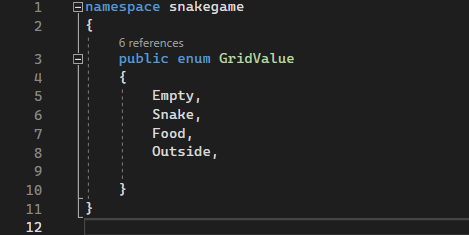
Az egyéni projektem egy kígyós játék lenne. A játék lényege, hogy a kígyót irányítva föl, le, balra, jobbra irányokba megetessük az ételével. Ha ez sikerült, a kígyó nőjön egy blokkot a pontszámunkkal együtt, és folytatódjon tovább a játék. Ha a kígyó fejével véletlenül neki megyünk a pálya szélének, a pontunk lenullázódik, és előröl kezdődik a játék.

Folyamata:

* A szükséges fájlokat létrehoztam a fejlesztői környezetben, illetve a játékhoz szükséges image fájlokat hozzáadtam az állományomhoz.

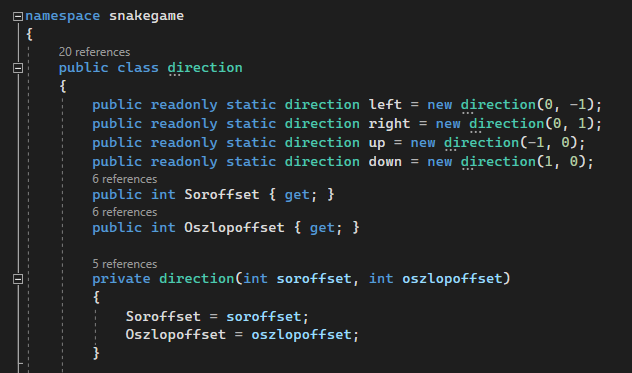
Rács szerkezete:

* A projekthez hozzáadtam GridValue nevű osztályt, ahol felsoroltam, hogy egy rácsban mi jelenhet meg. Egy üres blokk, a kígyó egy része, a tápláléka és egy úgymond kimeneti értéket jeleníthet meg a rácsunk. Utóbbira azért lesz szükség, mikor a kígyó neki ütközik a pálya szélének.

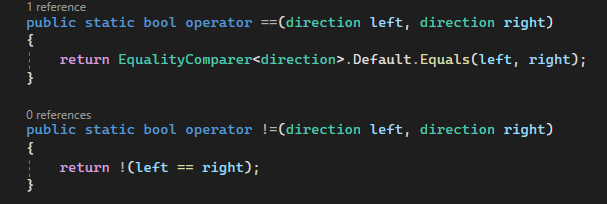


Irányok:

* A következő osztály amire szükségem volt az irányokat foglalja magába, így tehát létrehoztam “directions” néven. A rács sorokra és oszlopokra van bontva, ám az irányok maradnak, föl, le, jobbra, balra amiket statikus változókként adtam hozzá. Így iránytól függően adtam neki egy értéket, vagy adjon hozzá egy sort/oszlopot, illetve vagy vegyen el belőle.



* Továbbá létrehoztam egy ellentétes metódust, aminek a lényege, hogy új irányt hozunk létre ellentétes sor- és/vagy oszlopeltolással.
* Ezután felül szeretném írni az egyenlőket, így legeneráltam egy hash kódot az egyszerűség kedvéért, amivel könnyedén tudom alkalmazni az irányokat.

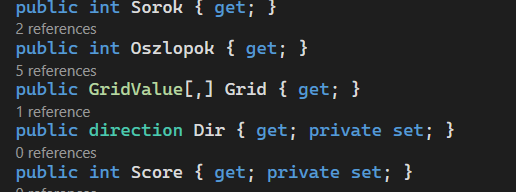


Aktuális pozíció a rácsban:

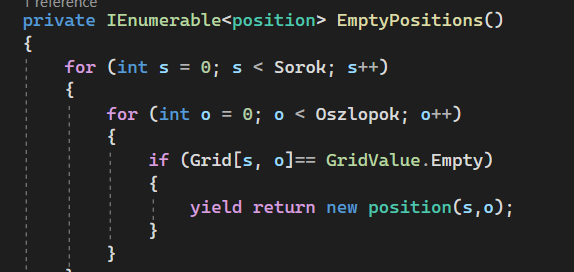
* Szükségem volt egy olyan osztályra is, amely egy pozíciót képvisel a rácsban. Az osztály tárolni fog egy sort és egy oszlopot, illetve azok tulajdonságait és értékeit mentse el. Létrehoztam egy olyan metódust, ahol azt a pozíciót fogja visszaadni, amelyet a lépésünkkel az adott irányba mozgatva kaptunk, és így egy új pozíciót fogunk kapni. Majd segítségemre itt is legeneráltam egy hash kódot.

A játék aktuális állapota:

* Létrehoztam az összes segítő osztályt, így most következik a játék aktuális állapota. Ezzel az osztállyal megalkottam a játék logikáját. Hozzáadtam néhány tulajdonságot, mint például a rácsban lévő sorok és oszlopok számát, kígyó iránya, a pontszám, illetve a rácsot is.



* A következő listámban tároltam a kígyó jelenlegi pozícióját, utána létrehoztam egy véletlenszerű objektumot, amire kifejezetten szükség lesz a programhoz.
* Szükségem volt egy olyan konstruktora, ami a rács sorait és oszlopait veszi alap paraméterként. Ebbe eltároltam ezeket a számokat a tulajdonságaikba.
* A következő metódusom az lesz, ami hozzáadja a kígyót a kívánt sorba és oszlopba, majd meghívtam az utóbbi konstruktorból.
* Szükségem van továbbá egy olyan metódusra is, amely visszaadja az üresen hagyott rácsokat. Végig nézzük For ciklusokkal az összes sort és oszlopot, hogy üres e az adott mező. Ha igen visszakérem ezt a pozíciót.



* Az ételhez létrehoztam egy listát az üres pozíciókról. Általános esetben egyszerűen visszatérünk, majd véletlenszerűen kiválasztunk egy üres pozíciót. Majd meghívtam a konstruktorban, ugyan úgy, mint a kígyót.

A játék még íródik...