Atestat a competenţelor profesionale

Clasa Matematică- Informatică

**Sorbán Előd**

XII. A.

**SPANzer**

Szakirányítók:

Szélyes Emőke

Godra Hajnal

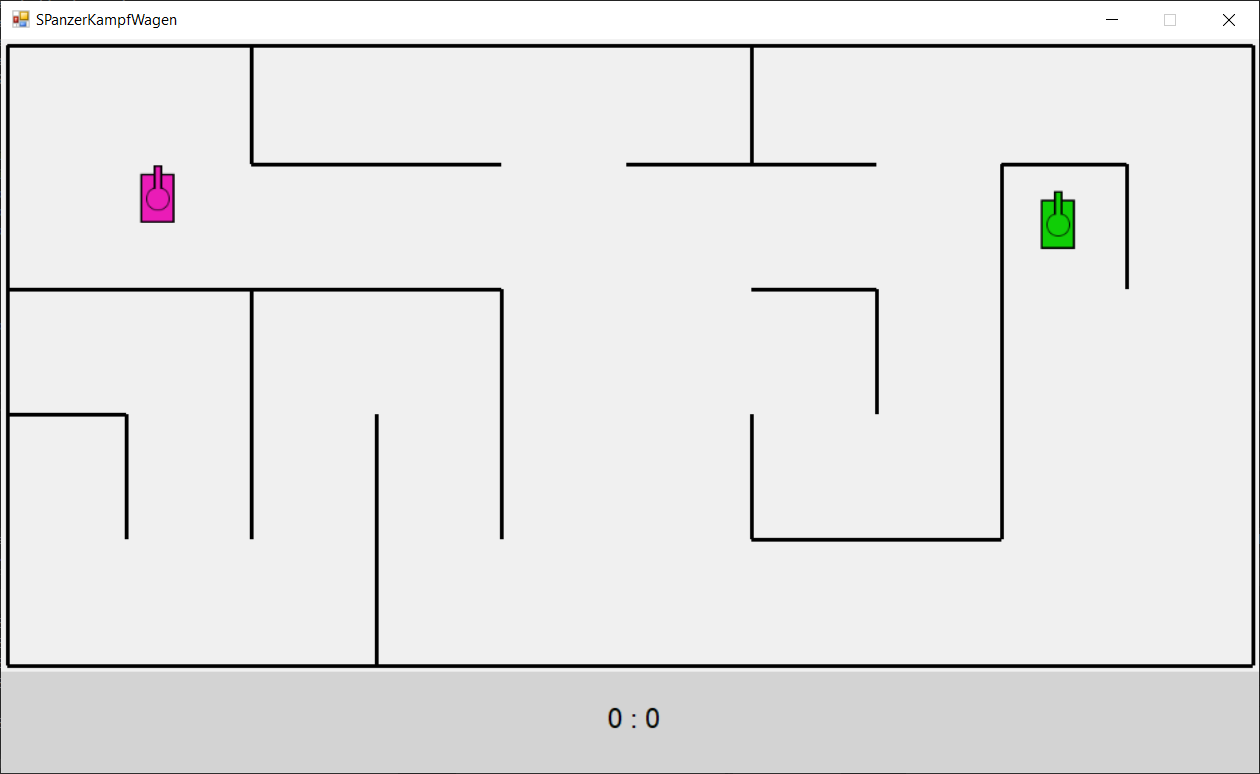
**- 2022 -**

**SPANzer**

Windows Forms App (.NET Framework 4.6.1)

Microsoft Visual Studio C# 2022

Multiplayer Arcade

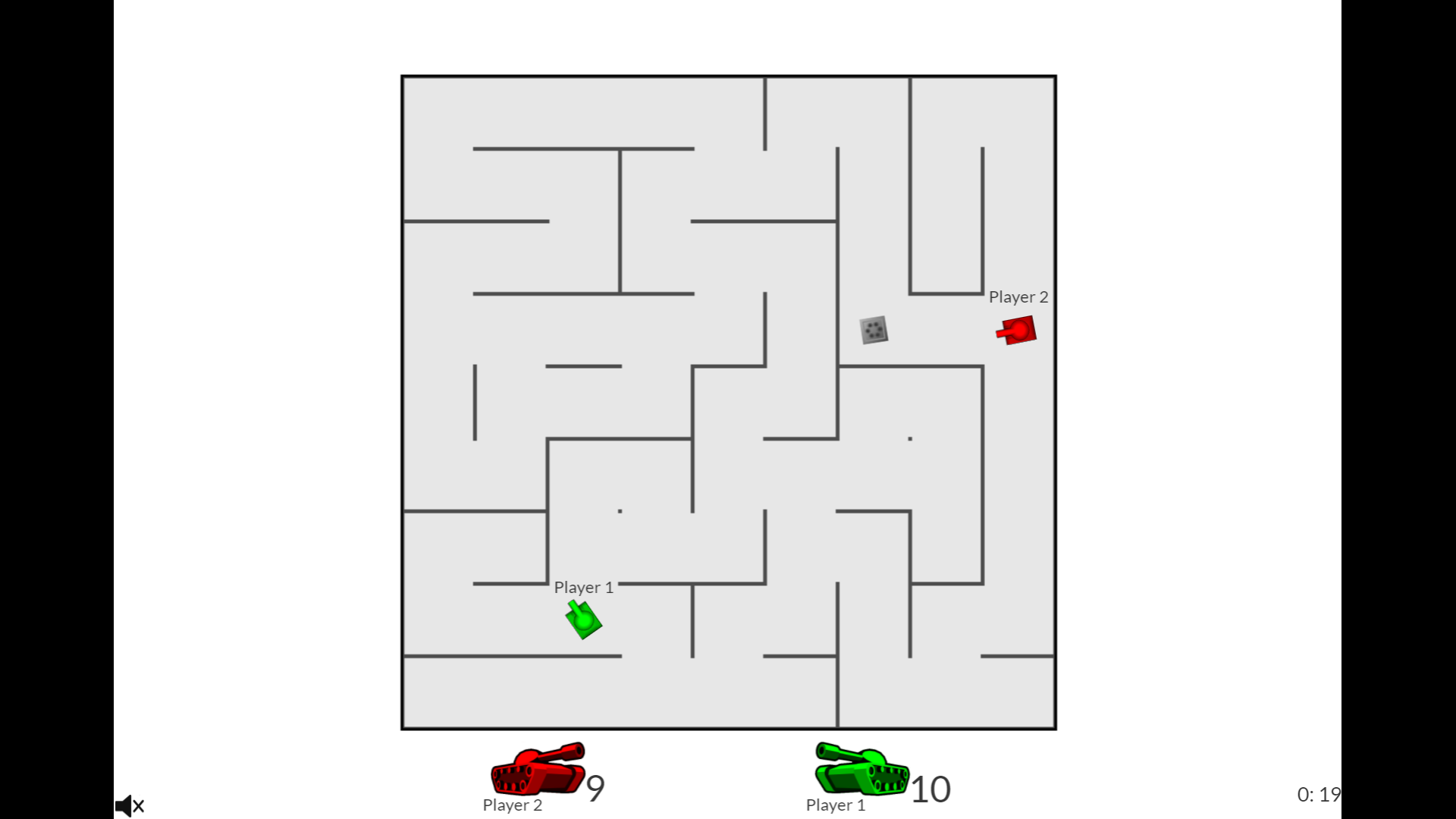


Tartalomjegyzék

* Ötletforrás 4
* Felhasznált fejlesztői eszközök 5
* A játék lényege 6
* A játék leírása 7
* Fejlesztési opciók 10
* Végszó 11

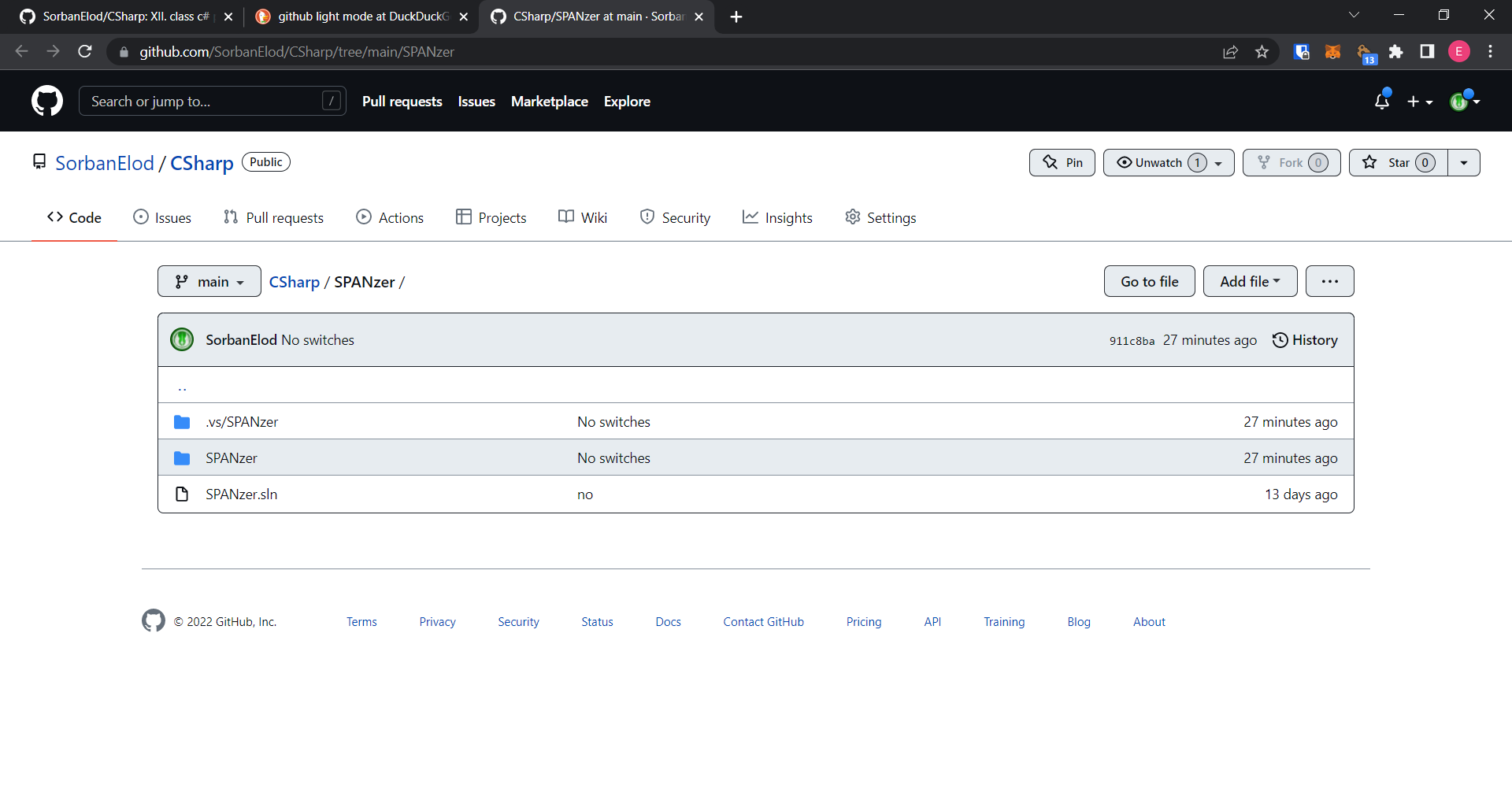
Ötletforrás

A 2007– ben megjelent Tank Trouble (AZ) nevű kétszemélyes online árkád–játékot írtam meg.



Felhasznált fejlesztői eszközök

* Visual Studio 2022
* GitHub
* GitHub Desktop
* Stack Overflow
* Notepad++

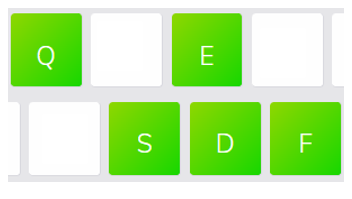


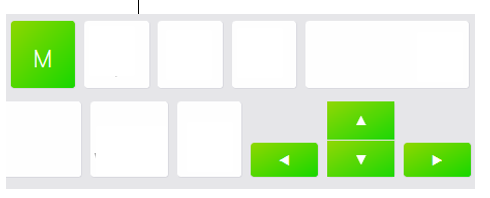
<https://github.com/SorbanElod/CSharp/tree/main/SPANzer>

***A játék lényege***

A játék egy kétszemélyes, felülnézetes árkád játék, amelyben két tankot irányítunk a billentyűzet segítségével.

A játék során a két játékos a billentyűzet segítségével navigál az előre megrajzolt pályákon, amelyek közül véletlenszerűen töltődik be az egyik.





Az irányítás relatív a tank helyzetéhez képest, tehát az előre gombbal a tank saját pozíciójához képest előre megy, ugyanígy a hátra gombbal a tank tolatni kezd.

A jobbra és balra gombok a tankot forgatják el a saját tengelye körül.

A játékosok golyókat lőhetnek ki egymásra, amelyik játékost találat éri az meghal, amelyik életben marad, az kap egy pontot, majd betöltődik egy új pálya és a játék folytatódik.

Egy játékosnak egyszerre öt tölténye lehet a pályán, kilövés után 10 másodperccel a golyó megsemmisül, ezután újra lőhet.

***A játék leírása***

A játék egy C#- ban íródott Windows Forms Applikáció, amely a. NET Keretrendszer 4.6.1-es verziójára épül.

A program 5 osztályt tartalmaz: *Tank.cs, Brick.cs, Walls.cs, Ball.cs, Bullet.cs.*

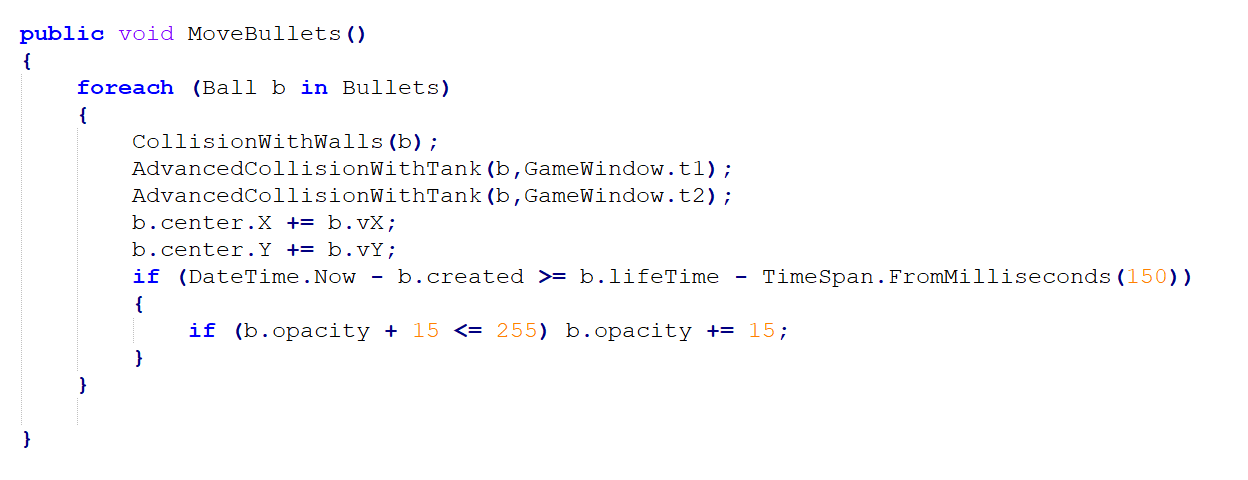
A *Tank* osztály tartalmazza a tank paramétereit (helyzet, file név, méret, irány, sebesség), és a tank mozgatására vonatkozó programokat, pl. az elforgatás, ütközés fallal, vagy ütközés egy másik tankkal. Az osztály két példánya automatikusan létrejön a játék elindításakor.

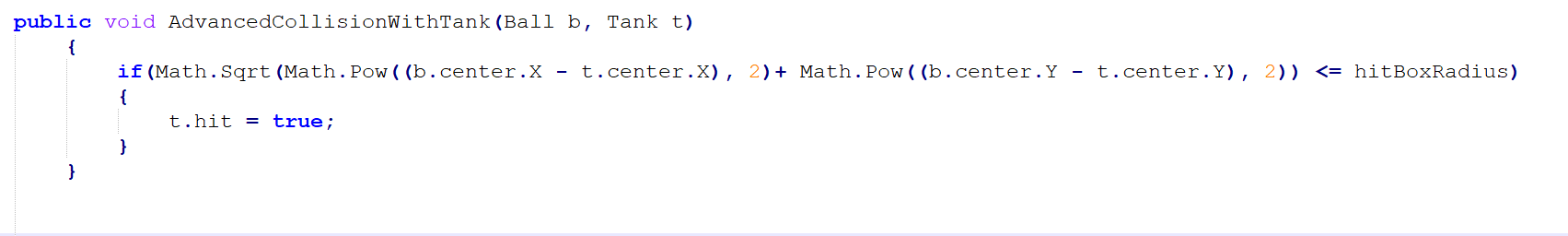
A *Brick* osztály tartalmazza a fal alkotóelemeinek tulajdonságait (szín, vastagság, kezdőpont, végpont és hogy a fal vízszintes, vagy függőleges).

A *Walls* osztályban egy lista tárolja a fal elemeit (*Brick* osztály példányai), amelyeknek egy szöveges fájlból olvassa be a paramétereit.

A *Ball* osztály tartalmazza a golyó tulajdonságait: sugár, sebesség, helyzet, szín, élettartam, és létrehozási idő.

A *Bullet* osztály egy sorban (Queue) tárolja a kilőtt golyókat, és figyeli, hogy ha egy golyó élettartama lejárt, akkor törli a pályáról. Itt van implementálva a golyó mozgatása és ütközése a falakkal, illetve a tankokkal való ütközése.



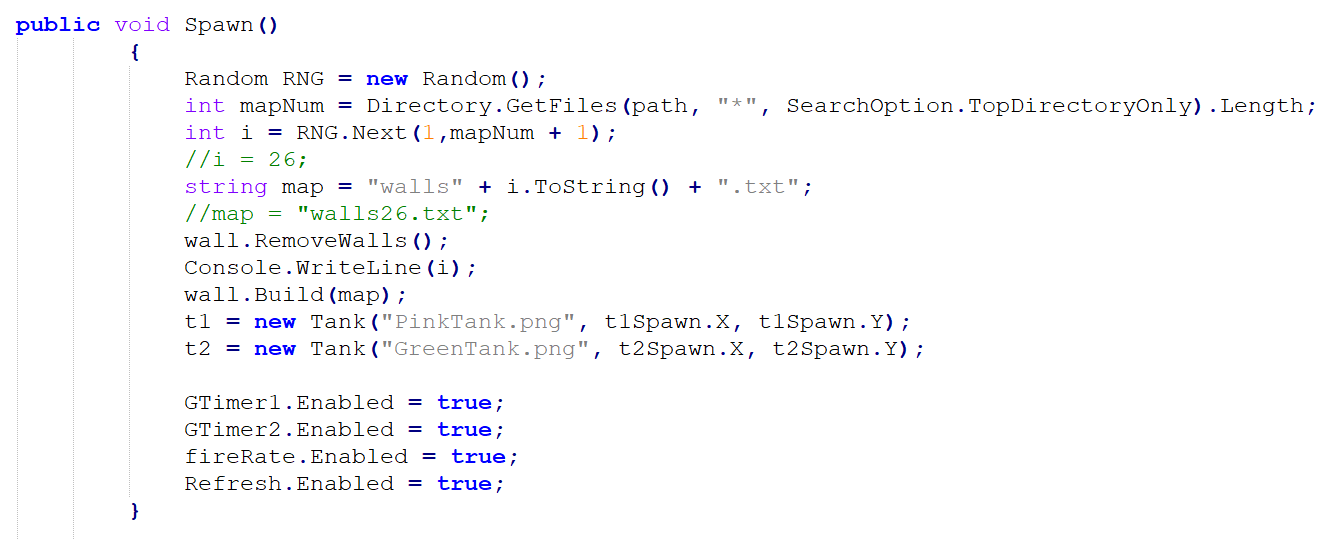


Jelenleg egy Form alkotja: a *GameWindow.cs.*

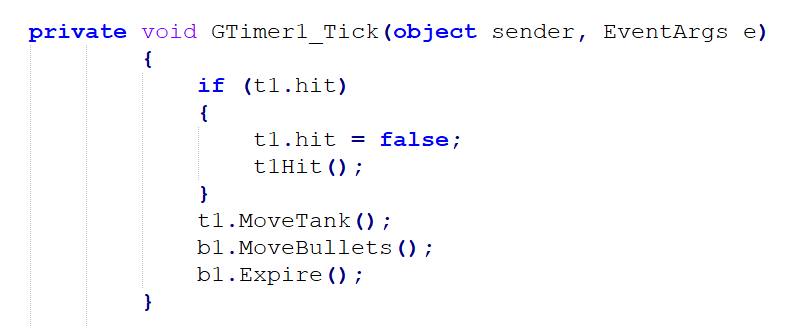
Az ablak két részből áll, egy *pictureBox*- ból, amely maga a játéktér és egy panelből, amelyen a pontszámláló található.

A program elindításakor egyből elindul a játék.

A *Spawn* alprogram feladata, hogy megszámolja hány pálya van, és ezek közül kiválaszt egyet véletlenszerűen, majd törli az előzőleg betöltött pályát (ha volt ilyen), ezután betölti az új pályát, elhelyezi a tankokat, és elindítja a timereket.



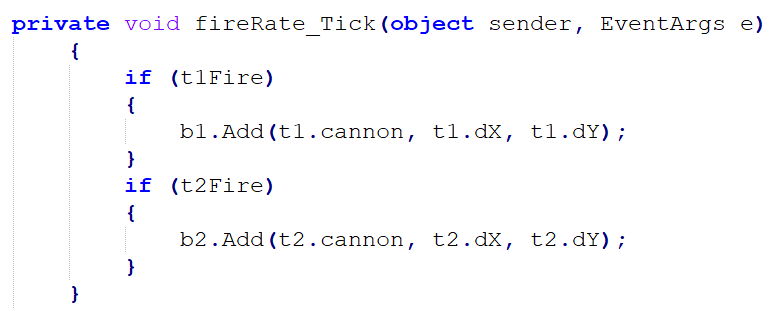
A *GTimer1* feladata, hogy minden ezredmásodpercenként megnéz, hogy az egyes játékos tankja el lett-e találva, majd ezután, ha nem volt találat, akkor mozgatja a tankot és a hozzá tartozó golyókat, illetve megnézi, hogy a golyók élettartama lejárt-e.

A *GTimer2* a második játékos tankját figyeli, és végzi ugyanezeket a műveleteket.

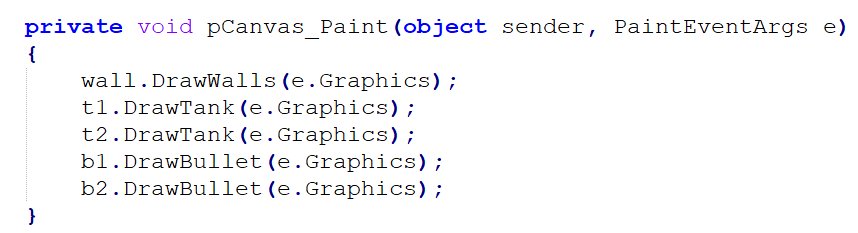
Azért fontos, hogy külön timer figyelje a két tankot, mert így jelentősen csökken a terhelés az adott végrehajtási szálon.

A Refresh timer feladata, hogy ezredmásodpercenként újrarajzolja a pályát.

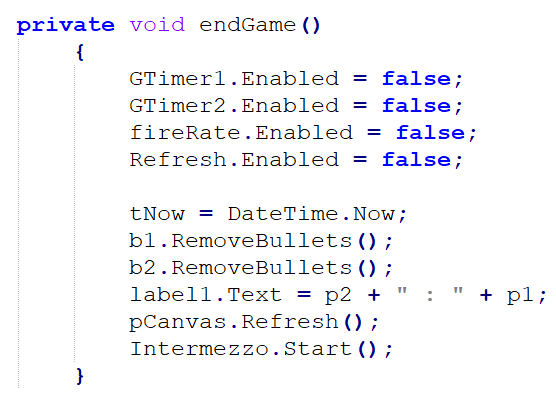
A *fireRate* timer tized másodpercenként ellenőrzi, hogy le van-e nyomva a lövés gomb, és ha igen, akkor kilő egy golyót.



A *pCanvas\_Paint* esemény meghívódik minden frissítésre (betöltéskor is), kirajzolja az az ablakra a pályát, a tankokat és a golyókat.



Amint az egyik tankot találat éri, a timerek leállnak, frissül a pontszám és egy másodperc várakozás után újra meghívódik a *Spawn* alprogram, és elölről kezdődik a játékciklus



***Fejlesztési opciók***

**Pálya generálása:**

A jelenleg előre megírt pályákat helyettesíteni lehetne egy pálya generálóval, ami nem kiválaszt egy előre megírtat, hanem kigenerál egyet.

**Játékmenü:**

El lehet készíteni a menüt, ami tartalmazna egy rövid útmutatót a játékhoz, egy beállítás menüt, ahol a játékosok kiválaszthatják a tank színét, a mozgás és fordulás sebességét.

A beállításokat lehetne tárolni egy szöveges fájlban.

**Különböző fegyverek:**

Az eredeti játékban a pályán véletlenszerűen különböző fegyverek jelennek meg (lézer, rakéta, stb.), ezeket felvéve a játékos előnyre tehet szert. Hosszú távon ezt a funkciót is szeretném implementálni.

***Végszó***

A SPANzer játékot elsősorban azért választottam, mert régen az egyik kedvenc játékom volt. Bár még közel sincs kész, büszke vagyok, hogy a magam erejéből ennyire sikerült megoldanom. A továbbiakban szeretném tovább fejleszteni. Remélem, elnyeri mások tetszését és örömmel fogják kipróbálni.

Jó szórakozást kívánok!