

Programozási technológia – 3. beadandó

Közös követelmények:

- A megvalósításnak, felhasználóbarátnak és könnyen kezelhetőnek kell lennie. Törekedni kell az objektumorientált megoldásra, de nem kötelező a többbétegű architektúra alkalmazása.
- A megjelenítéshez első sorban elemi grafikát kell használni. Az így kirajzolt 'sprite'-ok közül a játékoshoz tartozót billentyűzet segítségével lehessen mozgatni a jelenleg is standard (WASD) billentyűzet kiosztásnak megfelelően. Egyéb funkciókhoz egérhez kapcsolódó esemény vezérlőket is implementálhattok.
- Amennyiben nem algoritmussal generáltatod a játékteret, úgy legalább 10 előre definiált játékteret készíts különböző fájlokban eltárolva. Ügyelj arra, hogy mind az algoritmussal generált játékok esetén, illetve az előre definiált esetekben is végig játszható legyen az adott terület.
- Minden feladathoz tartozik egy időzítő, mely a játék kezdetétől eltelt időt mutatja.
- A dokumentációnak tartalmaznia kell a választott feladat leírását, elemzését, a program szerkezetének leírását (UML osztálydiagrammal), egy implementációs fejezetet a kiválasztott játék szempontjából és/vagy az általad érdekesebbnek gondolt algoritmusok leírásával. (Például: pálya generáláshoz implementált algoritmus.), valamint az esemény-eseménykezelő párosításokat és a tevékenység rövid leírását.
- A feladatleírás a minimális követelményeket tartalmazza. A játékok tetszőlegesen bővíthetők.

Feladatok:

1. Maci Laci (Yogi Bear)

A meséből jól ismert Maci Laci bőrbe bújva a Yellowstone Nemzeti Park megmászhatatlan hegyei és fái között szeretnének begyűjteni az összes rendelkezésre álló piknik kosarat. Az átjárhatatlan akadályok mellett Yogi élelem szerzését vadőrök nehezítik, akik vízszintesen vagy függőlegesen járőröznek a parkban. Amennyiben Yogi egy egység távolságon belül a vadőr látószögébe kerül, úgy elveszít egy élet pontot. (Az egység meghatározása rád van bízva, de legalább a Yogi sprite-od szélessége legyen.) Ha a 3 élet pontja még nem fogyott el, úgy a park bejáratához kerül, ahonnan indult.

A kalandozás során, számon tartjuk, hogy hány piknik kosarat sikerült összegyűjtenie Lacinak. Amennyiben egy pályán sikerül összegyűjteni az összes kosarat, úgy töltünk be, vagy generálunk egy új játékteret. Abban az esetben, ha elveszítjük a 3 élet pontunkat, úgy jelenjen meg egy felugró ablak, melyben a nevüket megadva el tudják menteni az aktuális eredményüket az adatbázisba. Legyen egy menüpont, ahol a 10 legjobb eredménnyel rendelkező játékost lehet megtekinteni, az elért pontszámukkal, továbbá lehessen bármikor új játékot indítani egy másik menüből.

4. Tron

Készítsünk programot, amellyel a Tronból ismert fény-motor párbajt játszhatjuk felülnézetből. Két játékos játszik egymás ellen egy-egy olyan motorral, amely fénycsíkot húz maga mögött a képernyőn. A motor minden másodpercben a legutoljára beállított irányba halad feltéve, hogy a játékos nem változtatja meg azt egy megfelelő billentyű lenyomásával. (WASD az első játékos, nyilak a második játékos.)

Az a játékos veszít, aki előbb neki ütközik a másik játékos fénycsíkjának vagy a képernyő szélének. A játék elején kérjük el a játékosok nevét és engedjük meg, hogy maguk válasszák ki a fényük színét. A játék végekor a győztes játékos eredményét növeljük meg az adatbázisban. Ha a játékos még nem található meg az adatbázisban, úgy szúrjunk be egy új sort. Egy menüpontban legyen lehetőségünk a 10 legjobb eredménnyel rendelkező játékost megtekinteni, az elért pontszámukkal, továbbá lehessen bármikor új játékot indítani egy másik menüből.

5. Saját játék

Saját játék ötlet kivitelezésére is lehetőség van.

Minimális követelmény, hogy az általad kitalált feladatra teljesüljenek a 'Közös követelmények'-ben leírtak, továbbá tartalmazzon a megoldásod legalább egy adatbázis táblát, melybe az alkalmazásod új sorokat szúr és az általad kitalált funkcionalitásnak vagy funkcionalításoknak megfelelően a szúrt sorok egy részét valamilyen szabály mentén visszaolvassa és megjeleníti.

Az általad kitalált/megvalósítani kívánt feladathoz tartozó leírást a beküldési határidő előtt el kell fogadnia az oktatónak! Tanácsos legalább 1 héttel a határidő előtt bemutatni a feladat leírást.