

Java programozás - 3. házi feladat

Labirintusos játék több játékoskal (100 p)

A feladat egy hálózaton keresztül játszható valósidőben játszható, labirintusos játék megvalósítása.

A játék lényege, hogy a játékos minél több pontot szerezzen különböző pontszerző objektumok begyűjtésével. Ezt a többi játékoskal versenyezve, egy labirintusban kell megtennie. Valós idejű, tehát mindenki bármikor léphet.

A labirintus négyzet alakú cellákból és a cellák közötti vízszintes és függőleges falakból áll, egy cellából a szomszédos cellák közül át lehet lépni azokba, ahol nem áll másik játékos, illetve a két cella között nincs fal. Szomszédos cella alatt élszomszédot értünk, tehát egy cellából legfeljebb négy irányba lehet lépni. Ha a játékos a szélső cellákról kifelé akar lépni (például az alsó sorban lefele), a hatás tetszőleges (toroid, szélén mindenhol fal, stb.). Az egész labirintus férjen ki a képernyőre (a lejjebb írt oldalsávval).

A szerver generálja a labirintust és küldje át a klienseknek. Elegendő, ha a szerver választ pár labirintus közül amit beolvas fájlból, majd valamilyen reprezentációban átküldi a klienseknek a labirintusba belépéskor.

A játékosokat jelző objektumok legyenek megkülönböztethetők! Ezeket a szerver és a kliens is tárolja egy-egy azonosítóval együtt, és a szerver a labirintusba való belépéskor véletlenszerűen osztja ki. Elég, ha a játékosok ugrásszerűen mozognak egyik cellából a másikba, de egy bizonyos időnek el kell telnie két ugrás között (minden játékosnál ugyanannyi).

A pontszerző objektumokat jelezzük körrel, legyenek megkülönböztethetők a játékosoktól. A több pontot adók nagyobbak legyenek. A pontszerzőket a szerver generálja véletlenszerűen időközönként, mennyiségben és pozícióra, de nem egymásra vagy olyan cellára, amelyen egy játékos van.

A játékot kliens-szerver architektúrában kell megvalósítani: a szerver vezérli a játékot, a kliensek csak a játékos mozgását küldik el a szervernek, illetve megkapják a játékban szereplő objektumok helyzetét: a játékosok és a pontok helyét akkor, amikor változik. A szerver figyel a lépések helyességére, vezeti a játékosok pontjait.

Egy játékost a neve alapján azonosítunk, csatlakozáskor ezt megadja, ha nem létezik ilyen név, akkor a szerver létrehozza 0 ponttal.

Egy szerverhez N játékos csatlakozhat. A csatlakozáskor a kliens kiírja a csatlakozott játékosok pontszámát, és lehetőséget ad a labirintusba való belépésre (ha már legalább 2 játékos csatlakozott).

A labirintusba való belépéskor a játékosok véletlen helyre kerülnek, azonban ez nem lehet olyan hely, ahol egy másik játékos vagy pontszerző objektum van. A mozgás a nyilakkal történjen, Esc-el lehessen kilépni a menübe. Egy oldalsó

sávon jelenjenek meg a benn tartózkodó játékosok az őket jelző ikonnal és a pontszámukkal.

A menübe kilépéskor a kapcsolat a szerver és a kliens között megszakad, erről a többi játékost is értesíteni kell. A játék folyamán az oldalsávnak követnie kell játékosok pontjait illetve újonnan belépett játékosokat is fel kell tüntetni (a kilépetteket pedig levenni).

A szerver legyen szálbiztos, és a GUI ne fagyjon.

JavaSlat a teszteléshez: írj olyan klienst, amelyik adott térképen előre beprogramozott mozgást végez, pl. jobbra-balra mozog.

JavaSlat a hálózati kommunikáció menetéhez:

1. Kliens a játékos nevével csatlakozik a szerverhez
2. Szerver elküldi a többi játékos nevét, mindenki pontszámát.
3. Kliens jelzi, hogy be szeretne lépni a labirintusba
4. Szerver elküldi a játékos ikonját (azonosító stringet), a másik klienst értesíti
5. A szerver folyamatosan küld üzeneteket az eseményekről, a kliens a felhasználói eseményről küld üzenetet
6. Ha a kliens kilép a labirintusból, a szerver szintén szól a többi kliensnek, és a többi játékost értesíti.

A kommunikációt meg lehet például szöveges üzenetekkel valósítani, úgy, mint:

Kliens: HELLO <játékos neve>

Szerver: OKE <játékos neve> PONTSZAM <játékos pontszáma> PARTNER1
<partner neve> PARTNER_PONTSZAM1 <partner pontszáma>

Hibajelzés például: HIBA Túl sok csatlakozott kliens

Pontozás

A hálózakezelés és a szerver kliens kapcsolt: 35p A GUI helyes, igényes kezelése: 55p szerializáció 10p

Plusz pontok

Animáció (5p)

A pontokat adó objektumokon történjen valamilyen animáció a játék során.

Leadási határidő: 2019. május 2. 07:59.