

Párhuzamos és eseményvezérelt programozás beágyazott rendszerekben

Házi feladat

Jedla Martin
Gazdag László

Feladat megfogalmazása

A feladatunk egy űrhajós játék megvalósítása volt, ami az STK3700 fejlesztőkártyán, és számítógépen, linuxon fut egyszerre. A két különböző platformon lévő program soros porton kommunikál.

A pálya egy tetszés szerint állítható méretű téglalap, amin akadályok vannak elhelyezve. A pálya széleit és az akadályokat is '#' karakter jelöli. A játék elindulásához az STK3700 fejlesztőkártya egyik nyomógombját kell megnyomni, a játék elindulása után a hajó a pálya bal oldalán jelenik meg. A hajó mindig a program által megadott időközönként lép, az irányát a nyomógombok segítségével lehet változtatni. A hajóval nem lehet visszafordulni és visszafele irányba menni. Ha a pálya tetejéhez vagy aljához ér, akkor azon továbblépve az ellentétes oldalon a következő lépés után megjelenik. A pálya végigvitele után új pálya generálódik, a hajó visszakerül a pálya elejére, és a lépésidő kevesebb lesz, azaz gyorsabban lép a hajó.

Linux rész

A linuxon futó programhoz mindösszesen két task kellett csak. Az egyik task bonyolítja le a soros porti kommunikációt, a másik task felel a kirajzolásért.

Soros portot kezelő task

A soros portot kezelő task először egy definiált START értéket vár. Ezután beolvassa az akadályok koordinátáit, majd a hajó koordinátáit olvassa be sorra. Amikor a pálya végéhez ér, akkor ezután ez a szekvencia újra kezdődik.

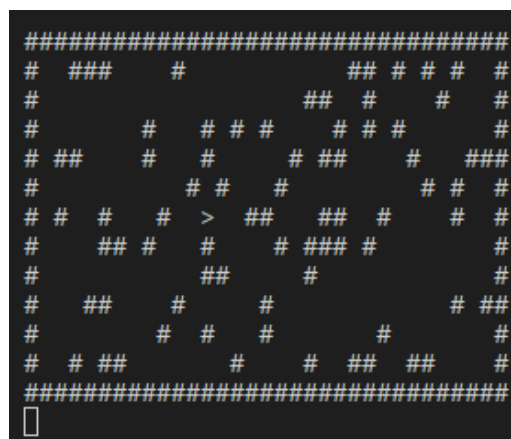
Minden hajókoordináta beolvasása előtt a saját szemaforát el akarja foglalni, illetve minden hajókoordináta beolvasása után felszabadítja a kirajzoló tasknak szánt szemaforot.

Kirajzoló task

A kirajzoló task mindig kirajzolja a pályát az akadályokkal, és a hajóval a koordináták szerint.

Mindig a kirajzolás előtt várja a saját szemaforját, és a végén felszabadítja a soros portot kezelő task szemaforját.

A játék megjelenése:



STK3700 rész

Az STK3700 felel a játék futtatásáért és soros porton kommunikál a linuxos programmal. A játék a kártya két nyomógombjával irányítható, a soros portra pedig a játék vezérléséhez szükséges értékeket és az akadályok, valamint a hajó koordinátáit küldi ki. A program elején define-okkal beállítható a pálya mérete és az akadályok száma, valamint a gyorsítás mértéke (ez az a szám, amivel a sebesség megszorozódik minden egyes pálya újrarajzolásnál).

Gombok kezelése

A gombok lenyomását megszakításon keresztül észleli a program. Mindkét gombhoz tartozik egy megszakítás kezelő függvény, ami egy-egy taskot indít el. Ezek a taskok a játék aktuális állapotának megfelelő műveletet hajtják végre: játék közben az irányt változtatják, a játék indítása előtt illetve a végén pedig alaphelyzetbe állítják a játékot és elindítják azt.

A hajó léptetése

A hajó léptetésének időzítését egy szoftveres időzítővel valósítottam meg. A hozzá tartozó callback függvény a játék futó állapotában elindítja a hajó kezelő taskot, ami elvégzi a hajó beállított iránynak megfelelő léptetését és szükség esetén a pálya újragenerálását.

Az akadályok kirajzolása

A megadott számú akadályt véletlenszerűen kell elhelyezni a pályán. A véletlen számok generálását a `rand()` függvénnyel végzem, amihez az `srand()` függvény seed paramétereként a rendszerindítás óta eltelt milliszekundumok számát kapja meg. A véletlen számok miatt elképzelhető, hogy két akadály koordinátái megegyeznek, ami kevesebb akadály megjelenését eredményezi. Ennek elkerüléséhez az akadályok koordinátáit a véletlen beállítás után ellenőrzöm és amíg megegyezik egy másik akadály koordinátaival, addig újra generálom.

Kommunikáció

A program számok küldésével kommunikál. Az indításhoz, az akadályok újrarajzolásához és a játék végéhez rögzített kódokat használ. A koordinátákat karakterként, ascii kódokban küldi. A 0 karakter kódja a 48, ezért minden számot ezzel az értékkel tol el. (Próbáltuk a koordináta értékével megegyező ascii kódokat küldeni, de az első tíz ascii kód speciális karakter, ami miatt virtuális géppel nem működött.)