

Sziasztok, Kiss Lőrinc vagyok, jelenleg másodéves. Engem is, mint másokat, a videojátékok világa vonzott be ide az egyetemre, és bár már rájöttem, hogy az informatikában mennyi más lehetőség rejlik ezen kívül, a játékok megalkotása még mindig nagy álmom.

Egy évvel ezelőtt nekem is Czirkos Zolival volt Programozás alapjai 1 tárgyam, és amikor megtudtam, hogy olyan házit csinálhatok, amelyet csak szeretnék, és a grafikus felület megalkotása nem csak hogy megengedett, de még tutorialok is vannak hozzá kapcsolódóan az InfoC-n, felcsillant a szemem: eljött az én időm.

Sajnos rajzolni nem igazán tudok, és a program mérete is elég korlátozott volt, amit le lehetett adni, ezért meg sem fordult a fejemben, hogy szép grafikát alkossak, és csakis az SDL primitív formáiból építkeztem.

Akkor van egy egyszerű játékunk primitív grafikával. Honnan ismerős ez? Egyből az arcade kabinetek ugrottak be, bár sajnos én az arcade fénykora után születtem.

Kezdetnek egy olyan játékot gondoltam ki, ahol egy kukásautóval járja a játékos a várost, és úgy gyűjti össze a szemetet a házak előtt. Végül inkább egy egyszerűbb és stilizáltabb grafika lett az eredmény.

Akkor a lényeg: a szabályok:

- A játékos egy színes koronggal mozog a négyzet alakú mezőkből álló pályán. A mezőkben kezdetben akad egy kevés memóriaszemét, amit a játékos a felette áthaladással eltüntethet.
- A játék akkor kezd izgalmas lenni, amikor megjelennek a "programok", amik bizonyos számú mezőt átfognak. Ezek halványan jelennek meg, és egyre erősebb színűek lesznek, egészen addig, amíg nem "aktiválódnak"
- Ekkor a játékos pontokat kap annak függvényében, hogy a program mezőin mennyi szemetet sikerült eltakarítania. Ilyenkor a program belsejébe nem tanácsos bemenni érthető okokból, ezért az ilyenért pontlevonást kap a játékos.
- Pár másodperc után a programok eltűnnek, és megjelenik alattuk a maguk után hagyott memóriaszemét, és kezdődik az egész előlről.
- A játék célja az idő lejártával minél több pontot összegyűjteni.

Alapjában véve ennyi lenne az egész játék. Nyilván ez túl egyszerű lenne egy nagyházinak, így arra gondoltam, hogy ennek egy szép keretet adhat, ha pár extra funkciót is megvalósítanék. Ezért került bele egy egészen rendesen navigálható menü, pár játékbeállítással, mint a pálya mérete és a nehézségi szintek, egy leaderboard, a játékos színének beállítása valamint játék közben is lehet szünetet tartani. Voltak más ötleteim is, mint egy tutorial, és ahogy arcade játékhöz illik, egy többjátékos mód, de az már nem fért bele a félévbe.

Most az előadásra készülve megint belenyúltam kicsit, hogy egy controllerrel is lehessen vezérelni. Így utólag megnézve rendkívül kaotikus az egész. Az egész program a kommenteket és az üres sorokat leszámítva is majdnem 1400 sor, és 24 globális változót használtam. Sajnos eléggé freestyle kezdtem bele az egészbe. Volt egy ötletem, leültem, és elkezdtem írni. A végére kialakult valami, de valószínűleg ha újra hasonló feladat előtt állnék, egy kis előre tervezéssel sokkal kevesebb idő alatt sokkal kevesebb sorban lehetne ugyanezt megoldani.

Az egész játék esemény-alapú, a képfrissítést is egy 15ms-ként generált esemény adja. Ezzel a játék 66FPS-el fut. Az animációkkal volt talán a legtöbb munka: a legelső változatban minden képfrissítésnél rárajzoltam az ablakra egy félig áttetsző fekete téglalapot. Ez jó ötletnek tűnt, mivel így mindig egyel halványabb és sötétebb volt a játékos előző állapota, és ez nagyon tetszett. Akkor

találtam ki azt is, hogy a menüben legyen egy pár játékos objektum kirakva, amik véletlenszerű színnel, mérettel és irányban mozognak a háttérben. Eleinte jó is volt, és tovább is léptem, de amikor egyszer teljes képernyőn indítottam el a játékot, annyira elkezdett akadni már a menü, hogy egy gombnyomásra való reakcióra is hosszú másodperceket kellett várni. Ahelyett, hogy lemondtam volna a háttérről, inkább újraírtam az egész kirajzolófüggvényt és a játékos adatstruktúráját. Ettől kezdve nem csak a játékos színét, méretét és helyét tárolta, hanem egy 15 elemből álló tömbben tartotta az játékos előző 15 pozícióját. Minden kirajzoláskor léptette egyet a tömb elemeit, mindegyik elemet egy kicsivel átlátszóbban, mint az előző. Az átlátszóságot eleinte egy lineáris függvény adta meg, de pár próbálkozás után rájöttem, hogy egy koszinuszfüggvénnyel sokkal jobban néz ki. A felhasználó beállíthatja a korong színét, a pályán a programok pedig egy majdnem ellentétes színnel fognak megjelenni a kontraszt érdekében.

A pályát egy dinamikus, 2 dimenziós karaktertömbben tárolja a játék, ami mellett egy másik hasonló tömb azt tárolja, hogy melyik mezők foglaltak jelenleg. Eleinte két előre lefoglalt tömböt használtam, de plusz funkciónak beleírtam a változtatható pályaméretet (adott keretek között), így muszáj volt átállni dinamikus tömbökre. Ez azért volt lényeges, mert a programokat véletlenszerűen generálja a játék, és semmiképp sem akartam, hogy a programok összecsússzanak. Eleinte összecsúszás esetén addig generálta újra az adott programot, amíg rá nem fért valahova a táblára, de ez időnként befagyásokat okozott, ezért végül úgy lett megoldva, hogy elkezd balra léptetni, ha elérte a tábla szélét, akkor lerakta a következő sor elejére, és ha elérte a pálya jobb alsó sarkát, akkor lekicsinyítette egyel a méretét, és elindította megint a bal felső sarokból. Ez egy jó darabig működött, de párszor ez nagyon lelassította az egész játékot, ezért végül hozzáadtam azt is, hogy egy bizonyos számú léptetés után inkább ne próbálkozzon tovább.

A menüben megjelenő gombokat és csúszkákat két dinamikus kezelt láncolt listában tárolom. Jelenleg már túl vagyok a prog2-n is, és be kell, hogy valljam, sosem gondoltam, hogy valamikor ennyire vágyni fogok egy heterogén kollekcióna. A kontroller beintegrálásánál külön hatalmas munka volt, hogy nem csak arra kellett figyelni, hogy melyik gomb van kijelölve, hanem ha a felhasználó épp egy csúszkához ért, akkor az A gomb megnyomásával ne a legutolsó kijelölt gomb funkcióját hozza elő. Végül két külön változó tárolja a kijelölt csúszkát és gombot, és abból az egyik mindig NULL. Az egérrel való irányítás is hasonló alapon működik, de ott kicsivel könnyebb volt, amíg feltételeztük, hogy a kurzor alatt nem lehet egyszerre csúszka és gomb is.

Amikor az utolsó vagy az utolsó előtti prog előadáson Zoli az állapotgépekről és a függvénypointerekről beszélt, én épp az előző este fejeztem be teljesen a házimat. Egy állapotváltozóban el volt mentve a jelenlegi állapot, a main függvényt pedig egy switch uralta, ami minden állapotra megmondta, hogy mit kell tenni a kirajzoláshoz. Bár külön függvényekben volt megírva az oldalak kirajzolása, az egész switch mégis kb 600 soros volt. Elhatároztam hogy nem hagyom ilyen terjengősen a házimat, mivel ezzel az állapotgépes megoldással pár sorban meg lehet oldani az egészet. A main függvényt majdnem a nullából írtam újra, és végül egy függvénytömbbel, és 2 plusz sorral sikerült megoldani ugyanazt, ami előtte kb a kód harmadát elfoglalta. Minden tömbelem a saját oldalát inicializáló függvény-pointert tárol, a mellette lévő szám pedig azt jelöli, hogy Esc megnyomása esetén melyik állapotba kell váltania. Minden inicializáló függvény átírja a tömb legelső elemét, hogy a program ezután a megfelelő kirajzoló függvényt hívja meg.

A dicsőségtábla elemei is egy saját struktúrában vannak, ami tárolja a nevet, pontszámot és a játékos színét is, így minden pontszámot olyan színnel lehet kiírni, ahogyan azt megszerezték. A legmagasabb pontszámokat a legutóbb használt színnel együtt két külön szövegfájlba menti el a játék minden egyes alkalommal, amikor valamelyik pontszám meg lett döntve.

Az egész játék elég értelmetlennek és nagyon egyszerűnek tűnik, de nem is az volt a célom, hogy valami maradandót alkossak, csak ki akartam próbálni, hogy képes vagyok-e egy ilyen projekttel megbirkózni, és a progházi erre tökéletes keretet adott. Rengeteget tanultam ezalatt a projekt alatt. Megismertem az esemény-alapú programozást és egy alap grafikus könyvtárat. Azóta számos arcade-szerű játékot találtam és szerettem meg, és ez a projekt meghozta a kedvemet, hogy egyszer egy összetettebb, komolyabb arcade-szerű játékot is összehozzak.