

ArrowGame

Az ArrowGame egy, az Asteroids nevű játékra nagyon hasonlító 2+player PvP (Player versus Player) Top-down nézőpontú lövöldözős játék.

Indítás

„[...@...] .../...\$./build/ArrowGame”

A program betöltéskor a „Config/graphics.cfg” nevezetű fájlt megkeresi, és az abban tartalmazott információk alapján létrehozza az ablakot és a render kontextust.

Graphics.cfg tartalma: [Szélesség (int)] [Magasság (int)] [Scaling (float)] [Fullscreen (0/1)]

Amennyiben a fájl hiányzik, a program generál egy „alapértelmezett” fájlt, esetlegesen gép-specifikus adatokkal.

A játékos textúra betöltése is itt történik meg, ezt a program az „Assets/Arrow.png” fájlból tölti be. Ennek hiányában a program hibakóddal kilép.

Menü

Ezek után a program betölti a főmenüt, és átadja a kezelést a főmenü loop-jának.

Menü irányítás: fel, le, Enter, Escape, Csúszka állítás: balra, jobbra, illetve az ezeknek a gomboknak megfelelő bemeneti konfigurációk (később részletezem)

A főmenüben a játék neve („ArrowGame”) és 4 választási lehetőség található.(továbbiakban gombok, menüpontok) Az első menüpont („Random Game”) választásával a játék generál magának egy konfigurációt, majd egyből beledob az akcióba.

A játékgenerálást egy úgynevezett random.gcgf fájl szabályozza, a program ezt is a Config mappában keresi, hiányát beépített alapértelmezésekből készült fájlal pótolja. A fájl egyes sora tartalmazza az éppen hozzá tartozó beállítás értékét, ha az egy értéket tartalmaz, amennyiben viszont kettőt, akkor az a random szám generátor alsó és felső hataraként lesz értelmezve.

A „Custom Game” menüpont két al-menüt (Environment, Weapon) tartalmaz, melyekben különféle csúszkák segítségével szabható személyre a játékelmény. A csúszkák sorrendje megegyezik a .gcfg kiterjesztésű fájlban található információk sorrendjével. Az itt található alapértelmezések a custom.gcgf fájlból kerülnek beolvasásra.

*.gcfg fájl tartalma (separator: enter):

- [Játékosok száma (int)]
- [Játékosok gyorsulása (min – max¹, vagy float)]
- [Játékosok fordulási gyorsulása (min – max, vagy float)]
- [Játéktér súrlódása (min – max, vagy float)]
- [Játékosok életpontjai (min - max, vagy float)]
- [Fegyver típusa (min - max, vagy uint)]

1 Min és max típusa megegyezik az egyváltozós típussal

- [Fegyver sebzése (min - max, vagy float)]
- [Lőszer sebessége (min - max, vagy float)]
- [Lőszer-visszatöltődés ideje (min - max, vagy float)]
- [Tár mérete (min - max, vagy float)]

Az „Options” menüpont 4 almenüt tartalmaz: „Graphics”, „Controls”, „Random game settings”, „Custom game settings”.

A „Graphics” almenüben 4 slider (Width, Height, Scaling², Fullscreen) és 2 gomb (Apply, Back) található.

A „Controls” almenüben először a játékos sorszámát kell kiválasztani, és ekkor egy következő menübe lép, ahol kiválaszthatjuk a bemenet típusát, illetve az egyes inputok által figyelt gombot, joystick-tengelyt, vagy billentyűt.

A „Random game settings”-ben a gcfg által definiált alsó és felső határok szűkíthetők (egyesével a maximum és minimum értékek)

A „Quit” menüpont választására a játék nemes egyszerűséggel kérdés nélkül kilép.

Játék

Amennyiben elkészült a játék konfigurációja, elkezdődik a játék inicializációja. Betöltődik az input konfiguráció, létrejönnek a játékosok által irányítható „karakterek”(Továbbiakban háromszögek), stb. A háromszögek pozíciója kezdéskor minden esetben random.

Ezek után a játék elkezd frissíteni és kirajzolni magát.

A játék egy fekete háttéren zajlik, semmilyen akadály, fal nem található a pályán, leszámítva a képernyő széleit, meg magukat a játékosokat.

Input config

Az inputok egy player[sorszám] (szögletes zárójel nélkül) nevű, .icfg kiterjesztésű fájlból kerülnek beolvasásra, a Config mappából. A fájl minden sorában 2, szóközzel elválasztott szám található, ezek közül az első a bemenet típusát (keyboard, controller button, controller axis (positive), controller axis (negative)), a második a megfelelő enumbeli (SDL_Scancode, SDL_GameControllerButton, SDL_GameControllerAxis) értéket.³

Az inputok sorrendje⁴, illetve célja:

- Előre / Gáz / Fel – A háromszög gyorsít
- Hátra / Fék / Le – A háromszög lassít
- Balra – A háromszög balra forogni kezd
- Jobbra – A háromszög jobbra forogni kezd
- Lövés / OK – A háromszög lő, ha a tár nem üres
- „Boost” – Nincs semmi hatása

2 A felbontás hány százaléka valójában a render context felbontása

3 Igen, a játék támogat kontrollereket

4 Így kerülnek beolvasásra

- Pause / Back – Megállítja a játékot

A fájl hiányában a játék a Config/default.icfg fájlt tölti be, ha ez is hiányzik, elkészíti .

Háromszög-fizika

A háromszögek a játék fizikája szerint körök ($r = 3,75\text{px}^5$). Akármilyen irányban haladnak, hat rájuk egy súrlódási erő (viszonylag kicsi), mely input hiányában folyamatosan lassítja a háromszöget. Két háromszög ütközése úgy történik, mint 2 10Ft-os esetében a valóságban, (nincs energia veszteség, talán ez az egy más) (nem tudom ez melyik fajta ütközés a fizikában, nem vagyok fizikus *még*)(Nem okoz sérülést).

A háromszögek a pálya széléről is visszapattannak, ekkor sincs energia veszteség (tehát például lehet stratégia gyors irányváltás eléréséhez a falnak menni).

A háromszögekre a lövedékek is hatással vannak, minden lövés a lövedék irányával ellentétes irányú, a golyó sebességével és súlyával arányos ellen-erővel hat a háromszögre. (golyó súlya: Háromszög súlyának (egység) ezred-része)

Ezenfelül minden találat erővel hat a háromszögre, hasonlóan a lövésnél, annyi különbséggel, hogy az erő iránya megegyezik a golyó haladási irányával.

A háromszög irányításakor a háromszög gyorsulás-vektorát befolyásoljuk. (frame-enként különbözik, pillanatnyi, kizárólag inputtól függ).

Lövedék-fizika

A lövedékekre a pálya súrlódása nem hat. Háromféle lövedék-típus van a játékban: átlagos, lézer és hőkövető. Az átlagos lövedékre nem hat semmilyen extra erő, a lézer egy egyenest vizsgál, és ha talál sebez, a hőkövető golyóra pedig folyamatosan hat egy erő, ami húzza az ellenfél felé.

Játék közben a Pause gomb megnyomására megjelenik a Pause menü, ami 3 menüpontot tartalmaz: „Continue”, „Restart”⁶, „Quit to Menu” (szerintem a nevük nagyon jól elmondja, hogy mit csinálnak).

Particle-k

A particle-k (részecskék) szerves részét képezik a játéknak. Ezek kis, egypixeles színes pöttyök, amik bizonyos események hatására jönnek létre.

Háromszögek gyorsításakor a háromszögek hátuljából „jönnek ki”, azt a hatást keltve, hogy a háromszög emiatt gyorsul

Ütközéskor az ütköző játékosok színeiben repülnek mindenhová.

5 1 px = 1 cm

6 A játék nem jegyzi meg a kezdőállást, helyette ugyanolyan alapbeállításokkal indít egy új játékot.

Lövedék találatkor az eltalált játékos színeiben repülnek, azt a hatást keltve, hogy a háromszögből egy darab leszakadt.

Win condition

A játékot a játékosok addig játsszák, ameddig vagy meg nem unják, vagy már csak egy játékos maradt életben. Győzelem esetén egy menüben a játék kiírja a győztes játékos sorszámát (pl.: „Player 1 wins!”), illetve 2 lehetőséget: „New game”, és „Quit to Menu”. A játék az egyes játékosok győzelmének számát nem követi.

Building

1. Build mappa létrehozása (ha még nincs)
2. `cd Build`
3. `cmake ..`
4. `make`
5. Kész.

Esetlegessen szükség lehet a projekt root mappájában levő CmakeLists.txt fájl szerkesztésére, ugyanis Debian és Arch alapú disztribúciók között megeshet, hogy az SDL include-olása más és más módon van megoldva. Emellett a játék nem windows-ra lett írva, ettől függetlenül igyekszem majd időt szánni a cross-platformosításra is, de ez nem a NHF része.