

Karting Game in Unity

– Kovács Boldizsár –

Multiplayer

2021.12.15.

A többszemélyes játék megvalósításához, egy viszonylag friss könyvtárat használok, a NetCode-ot. Ez az MLAPI-hoz nagyon hasonló. Annyira, hogy a van egy projektkonverter, ami egy projektet MLAPI-ról Netcode-ra visz át.

Mivel kezdetleges, ezért sok probléma adódott vele, melyek voltak kritikusak is, és apróságok is. Legrosszabb, ami miatt csúsztam is egy keveset a leadással, hogy nem ismerte fel a betöltött Scene-t. Ezt egy jó időbe telt javítani, de még most is van benne sok kiküszöbölhetetlen probléma. Ezeknek a javítására van terv a jövőre nézve (1.1-es verzióba, ami majd kijön valamikor), de ezen hibákkal most kellett mit kezdenem, így a legjobbat próbáltam kihozni.

A kódom nagyrésze az, amin a többiek dolgoztak. Ezeket kellett összehangolnom egy olyan „Scene”-be, melyben mind ezt több gép között is lefolyhat. Ez viszonylag sikerült, de sajnos rengeteg megkötéssel kellett együtt élnem.

Lényegében kreáltam egy kisebb Lobby-t a stadion közepére. Ide viszonylag bárhány játékos tud csatlakozni. Ezt követően a Host el tudja indítani a játékot. Ezt úgy teszi meg, hogy mindenkit áthelyez a startvonalra egymás mellé (olyan 2-6 játékosnál férnek csak egymás mellett el alkalmasabbnak mondható módon). Ezzel együtt a pályát szinkronizálja a

klienseknél. Frissíti a pályát, a falakat, az ugratókat és a powerUp-okat is.

Ezek minden kliensnek lokálisan vannak csak meg. Szervertől egyszer megkapja a pályát, és onnantól elengedi a szerver a kliens kezét. Két okból, egy mert statikus a pálya, nincs változás játék közben. Továbbá azért is, mert a szerveren fut a fizika, ezért a kliens pályája csak iránymutató, hogy lássuk merre is van az előrefele. (program felől nézve ez a helyzet, de a játékos szeméből sokkal de sokkal fontosabb hogy lássuk a pályát, azaz a kliens oldal)

Az összehangolás főképp kliens oldalon van megoldva. A szerverre felküldött adatok a játékosok inputjai. E szerint vezeti le a szerver a játékot, legfőképp a fizikát, majd a kliensek erre hangolódnak rá. A klienseknél el van tárolva az aktuális fizikai állapot, ebből tudnak dolgozni, amíg nem kapnak újabb képet a szerver állapotáról, így addig is halad előre a játék folyamatosan náluk. Mikor megkapja a szervertől a következő állapotot, akkor a különbséget interpolálja. Ezzel folyamatos képet kapunk az eseményekről, és nem rontja a játékélményt szaggatásokkal.

Lehet szervert, host-ot és klienst kinyerni a programból, ezek a program indítása után rögtön kiválaszthatóak. Meg lehet adni egy IP címet is, melyre ha kliens lesz a program, fel próbál csatlakozni. Ha a szerver kíván egy kliensnek adatot küldeni, akkor azt a ClientRpc függvényekkel teheti meg, fordított esetben a ServerRpc-vel oldható meg ez. (általánosságban)

Egyéb projektbeállításokat nem eszközöltem, Asset-eket nem importáltam. A hálózatot leszámítva azzal dolgoztam csak, amivel a többiek dolgoztak.