# Szerver kommunikáció

A két instance közötti kommunikáció TCP alapú. A szerver – amennyiben nem használta – megnyitva a 7555-ös socketet, amelyeln keresztül tudja fogadni a kliens féltől az adatokat.

A kommunikációt megvalósítja egy Net nevű abstract osztály, melyből a Client és a Server leszármazik. A közös függvények, Network mindketten *override*-olnak a sendGameState és a connect és a disconnect. A connect és a disconnect végzi az össze illetve szétkapcsolást. A Server és a Client külön szálon futnak.

## GameState osztály

A Server és a Client közötti kommunikáció egyetlen egy (Serializable) osztály segítségével zajlik. Az osztálynak publikus változó vannak, viszont csak lokálisan egy-egy függvényben hozunk csak létre. A GameState osztály tartalmaz minden információt, mely a két isntance szinkronizálódásához szükséges. Tartalmazza a területeket információját (*Vector<Territory>*), az egyes játékosokról szóló információkat (*Vector<Player> players*), valamint azt, hogy milyen állapotban van az aktuális játék (*state*).

A GameState osztály az alábbi öt esetben kerül létrehozásra:

1. Kör vége gomb megnyomása. Ilyenkor frissíti a területeket, valamint átállítja az aktuális játékost, illetve frissíti a játékoshoz tartozó területek listáját
2. Ha megtámadták az adott játékost: ilyenkor a támadó elküldi a védekező félnek a dobása eredményét, és a védekező előtt felugrik az ablak megjeleik, melyben figyelmeztetik, hogy megtámadták, és lehetősége van dobni
3. A védekező elküldi a saját eredményeit a támadónak
4. A támadó játékos befejezi a támadást, és az eredmény kiértékelésre kerül
5. Szinkronizáció: A kliens elküldi az információját a szervernek, és a szerver a kliensnek. A játékosnév és a szín kiemelt fontosságú

## GameState feldolgozása

A GameState osztály feldolgozását a *Motor* függvényei végzik. Minden egyes esthez külön kiértékelés tartozik, mely frissíti a saját adatait a kapott adatok alapján.