

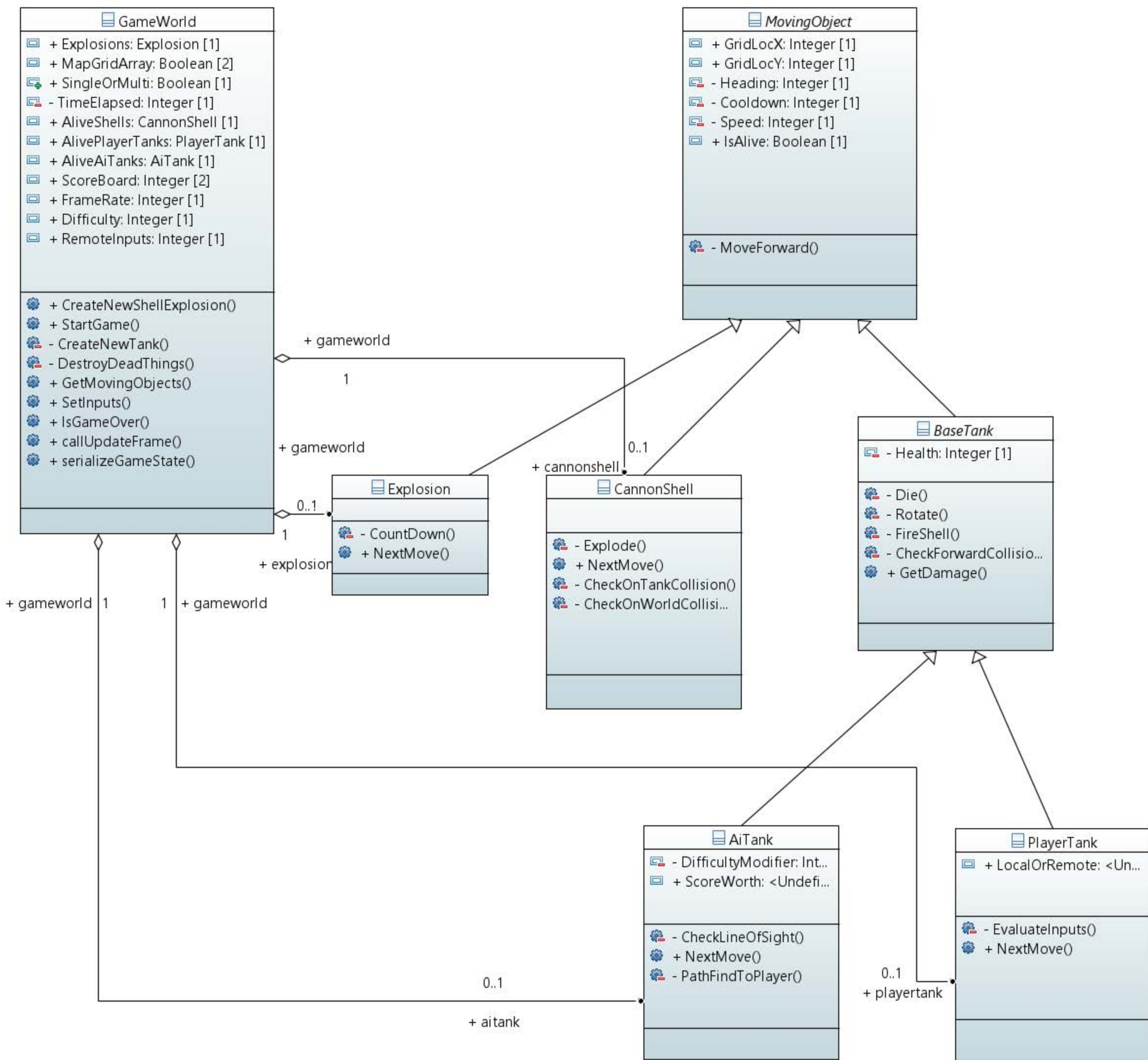
7. Csoport

Feladat felosztása:

GUI - Bauer Péter

Network - Kardos Bálint

Játék logika - Murányi Péter



GameWorld : A játékot vezérlő osztály, mely tartalmazza a játéklogika összes aktív elemét.

Fontosabb paraméterek:

- MapGridArray : A pályát tartalmazó 2D boolean lista, ahol True falat jelent, False pedig járható üres teret. Ezt használja az ellenséges tank útkeresője, az összes elem ütközéskeresője, illetve a GUI megjelenítési célokra.
- SingleOrMulti : A játékmódot tartalmazza, mindenki láthatja, ugyanis a létrehozott objektumok megegyeznek játékmódtól függetlenül, futásidőben ennek a változónak a függvényébe módosulnak az elemek funkciói.
- AliveShells, AlivePlayerTanks, AliveAiTanks : az aktív elemeket tartalmazza, különböző objektumok hozzáférhetnek olvasásra hogy pl, kiderítsék kit találtak el

Fontosabb metódusok:

- StartGame : GUI hívja meg, felinitializál mindent és elindítja a játékot
- callUpdateFrame : A játék logika és renderelés képkockákra van osztva, periodikusan futnak le a játéklogikát kezelő függvények, ez a függvény minden új képkocka elején fut le és kezeli az adott ciklusban elvégzendő dolgokat. Meghívja az összes aktív elem NextMove függvényét.
- CreateNewShellExplosion: tank vagy lövedék hívja meg, egy új lövedéket vagy robbanást hoz létre megadott paraméterekkel
- GetMovingObjects: összeszedi a GUI-nak az aktív mozgó elemeket fajtájukkal, orientációjukkal, és helyükkel a kirajzolásukhoz.
- CreateNewTank: játékmódtól függően létrehoz egy új tankot a kilőtt helyére, egy véletlen valid ponton a pályán.

MovingObject : absztrakt őssosztály, az összes aktív elemnek a játéktérben

Fontosabb paraméterek:

- Cooldown/Speed : A mozgó elemek mozgásának szabályozása úgy történik hogy minden valid mozgás/lövés után egy megadott számú frame idejéig nem mozoghatnak. A Speed megadja hogy a mozgások között mennyi várakozásnak kell történnie, a cooldown tartalmazza hogy még hány frame idejéig kell várakozni a következő lépésig, és minden frame ciklusban eggyel csökken az értéke
- IsAlive : boolean ami alapján a GameWorld tudja hogy egy halott elemről van szó amit meg lehet semmisíteni.

BaseTank : Tankok absztrakt őssosztálya

Fontosabb metódusok:

- CheckForwardCollision : Ellenőrzi hogy a tervezett mozgás valid pozícióba történik-e, nem-e ütközik tankokkal vagy fallal. A GameWorld map és aktív elemeket tartalmazó listáját használja
- GetDamage : lövedék hívja meg, levon az eltalált tank életéből
- FireShell : Lövedéket hoz létre adott paraméterekkel, amihez a GameWorld CreateNewShellExplosion függvényét hívja meg

PlayerTank: Játékos által vezérelt tank ami lehet helyi, vagy távoli játékos.

Fontosabb metódusok:

- NextMove: Új ciklusban lefutó függvény ami kezeli a viselkedést az adott ciklusban. Itt kapja meg a GUI vagy a távoli játékostól származó billentyű lenyomásokat. (megjegyzés: a NextMove minden MovingObjectben szerepel, de mindenhol eltérő az adott osztálytól függő funkcionalitást valósít meg)
- EvaluateInputs: Ha több billentyű is lenyomásra került az előző ciklusban, eldönti melyiket hajtja végre. Mozdásnak prioritása van a lövés felett.

AiTank: Ellenséges, számítógép által vezérelt tank, amely a játékost akarja megsemmisíteni (csak egyjátékos módban példányosul, a GameWorld hozza létre, véletlen helyre, ha épp nincs aktuális életben lévő ellenséges tank a pályán.

Fontosabb paraméterek:

- DifficultyModifier: Játék nehézségétől függően a tank máshogy viselkedik (lassabban mozog, tovább várakozik lövés leadása előtt), ezek befolyásoló paraméterei.
- ScoreWorth: Hány pontot kap a játékos a kilövéséért.

Fontosabb metódusok:

- NextMove: A tank ciklusonként lefutó vezérlő logikája. Számítógép vezérelt tank mindig először ellenőrzi hogy szabad rálátása van-e a játékosra, ha igen lövést ad le, ha nem a játékos felé mozog a legrövidebb úton.

CannonShell : lövedék osztálya, GameWorld tartalmazza, a Tankok hozathatják létre lövéskor

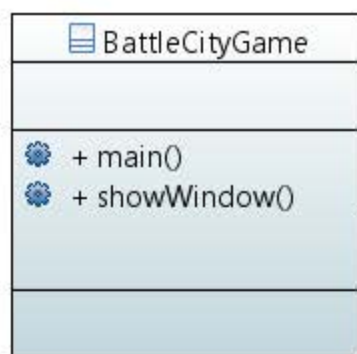
Fontosabb metódusok:

- NextMove : Lövedék logikája, GameWorld hívja meg minden ciklusban, eldönti a lövedék következő lépését
- CheckOnTankCollision / CheckOnWorldCollision : Tankokkal ellentétben azt vizsgálja hogy adott pozícióján van-e bármi, ugyanis nem az objektumok előtt robban fel hanem felettük.

Explosion: Lövedék által létrehozott osztály, grafikai célt szolgál, nem tud mozogni (egyszerűség kedvéért a MovingObject-ből lett leszármaztatva)

Fontosabb metódusok:

- Countdown / NextMove : a robbanás animációja adott ideig van jelen a pályán, a cooldown-t használja ennek meghatározására, a Countdown halottnak jelöli ha a cooldown eléri a nullát.



+ battlecitygame

1

0..1

+ GUI



+ tcpserver

1

1

+ gui

1

+ gui

1

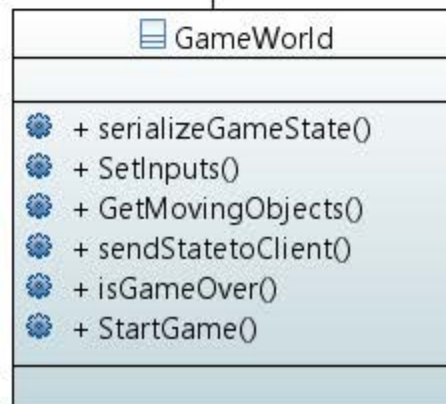
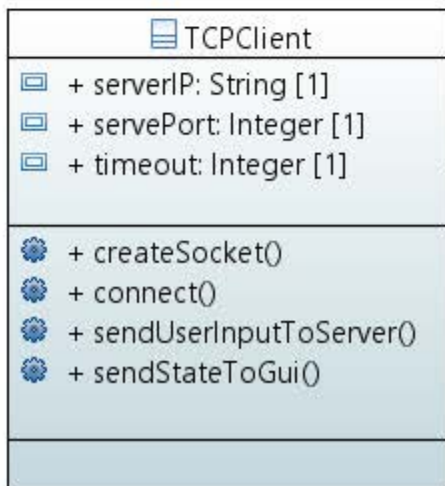
+ gui

+ gameworld

1

+ tcpclient

1



GUI: A játék megjelenítéséért felelős osztály. (Menü és a játék elemek kirajzolása.)

Fontosabb paraméterek:

- mapGridArray : A pályát tartalmazó 2D boolean lista, ahol True falat jelent, False pedig járható üres teret. Ezt használja az ellenséges tank útkeresője, az összes elem ütközéskeresője, illetve a GUI megjelenítési célokra.
- gameMode: A menün keresztül kiválasztott játékmódot tartalmazza: singleplayer, multiplayer host user, multiplayer remote user.
- difficulty: Szintén a menün keresztül kiválasztott nehézséget tartalmazza: easy, medium, hard.
- currentGameState: Az éppen aktuális játék állapotát tartalmazza, minden kirajzolandó objektummal (tankok, lövedékek, robbanás) és azok tulajdonságaival. Ezt az állapotot rajzolja ki a játék során a GUI és ezt frissíti a GameWorld is egy újabb frame kiszámítása után.
- scoreBoard: Ez tartalmazza a játékos(-ok) megszerzett pontjait a játék során.

Fontosabb metódusok:

- initComponents: Inicializálja a játékot a menüben kiválasztott opciók alapján: GameWorld, TCPServer, TCPClient. Elindítja a hálózatért és a logikáért felelős szálakat.
- showMenu: A menürendszer kirajzolásáért felel
- startGame: Elindítja a játékot és vele a játék kirajzolását is.
- drawGameFrame: Megrajzolja a pályát.
- keyboardActionPerformed: Billentyű lenyomásakor meghívódó metódus, mely letárolja azt egy FIFO szerű állományban.
- updateObjects: Frissíti a currentGameState alapján az objektumokat a pályán.
- deserializeGameState: Kliens oldalon visszaállítja a sorosított gameState objektumot és ez alapján rajzolja ki a kliens oldalon a játékot.
- sendUserInput: A lenyomott billentyűket tartalmazó állományt küldi tovább a GameWorld-nek feldolgozásra.

TCPServer: Multiplayer esetén a host oldalon indított szervert tartalmazza.

Fontosabb paraméterek:

- myIP : A szerver IP-jét tartalmazza.
- myPort: A szerver portját tartalmazza. (fix)

Fontosabb metódusok:

- sendStatetoClient: Kiküldi a kliensnek az éppen aktuális gameState-et, amit a GameWorld ad neki.
- sendInputsToGW: Tovább küldi a kliens felől érkező lenyomott billentyű adatokat a GameWorldnek.
- waitforClient: Várakozik egy, hogy kapcsolódjon hozzá egy kliens és megkezdődhessen a játék.

- getMyIP: Lekéri a gép IP-jét, hogy a kliens user tudja majd hova kapcsolódjon.

TCPClient: A játék megjelenítéséért felelős osztály. (Menü és a játék elemek kirajzolása.)

Fontosabb paraméterek:

- serverIP: Szerver IP, amit kézzel kell beírni a menüben.
- servePort: Szerver port. (fix)
- timeout: Az az időintervallum, ami után abbahagyja a szerverhez való próbálkozást a kliens és timeout hibát dob.

Fontosabb metódusok:

- sendUserInputToServer: A felhasználó által lenyomott gombokat tartalmazó listát, amit a kliens GUI adott át neki, tovább küldi a szerver felé.
- sendStateToGui: A szervertől kapott sorosított gameState-et küldi tovább a GUI-nak feldolgozásra.

BattlecityGame: A program indításkori belépő pontját tartalmazza.

Fontosabb paraméterek:

Fontosabb metódusok:

- showWindow: Inicializálja a GUI-t és elindítja annak a menüjét.

Multiplayer Scenario

