Beágyazott rendszerek szoftvertechnológiája

Házi feladat tervdokumentáció

*2022 Tavasz*

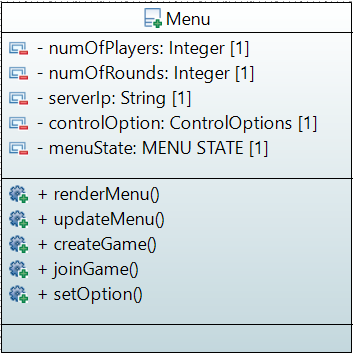
**KURVE FÍVÖR**

**Készítők: Hantos Márton, Harsányi Dániel, Fritsch Balázs**

**Konzulens: Naszály Gábor**

A program működését és a benne található osztályokat, illetve ezek kapcsolatait az UML diagram vizualizálja. A következőkben röviden bemutatjuk a főbb osztályokat, és ezeken keresztül a program tervezett működését.

## Menü osztály

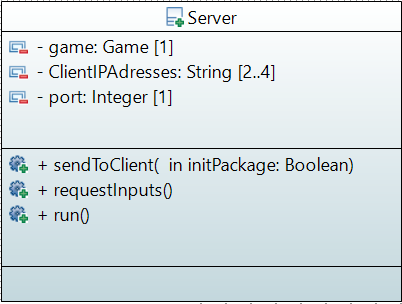


A menü osztály az az osztály a játékban, ami az alkalmazást, mint állapotgépet kezeli. A menünek három menüpontja van, ezek a Create Game, a Join Game és a Settings. A menüpontok beállításai a főmenü oldalán vannak, így csak egy menü oldal található az alkalmazásban. Create Game üzemmódban az adott játékos szerver üzemmódban, a Join Game pedig kliens üzemmódban indítja el a játékot. A Settings menüpont alatt kiválasztható a kanyarodás módja (vagy az A és D gombokkal, vagy pedig a nyilakkal kanyarodik a játékos).

## Client osztály

A kliens osztály a játékoshoz tartozó adatokat és render függvényeket foglalja magába. Ez az osztály valósítja meg a szerverrel való kapcsolatot. A kliens minden ütemben egyrészt elküldi a szervernek az aktuális irányítási állapotot (jobb-, bal kanyar vagy egyenes), illetve megkapja a saját pálya osztályán belüli változásokat a szervertől, amik alapján frissíti a pályát és a pontszámokat. MVC architektúrában a view és control feladatokat látja el.

## Szerver osztály



A szerver osztály a kliensből származtatott osztály, játékos szemszögből egy olyan kliens, amelyik a számításokat is elvégzi, tehát az MVC architektúrában a control és view feladatokon felül a modellezést is ellátja.

## Player osztály

A Player osztály fogja össze azokat az adattagokat, amikre a kliens oldali játékosnak szüksége van. Tehát az adott játékos tulajdonságát (a nevet, amit saját magának állít be), illetve azt az aktuális mozgatási állapotot, amit a billentyűzetről olvas be, illetve a hozzá tartozó egyedi azonosítót.

## ServerSidePlayer osztály

A Player osztály kibővített verziója, amire a szerver oldalon van szükség. A ServerSidePlayer osztály tartalmazza azokat az adattagokat, amiket a program a játéklogika alapján tárolni-, illetve számolni fog. A játékosok mozgatását is ezek alapján az adattagok alapján ennek az osztálynak a tagfüggvénye fogja elvégezni.

## Vector2D, CurvePoint, Curve osztályok

A fenti osztályok öröklési viszonyban vannak. A Vector2D az euklidészi térbeli pozíciót tárol, a CurvePoint ennek egy színnel történő kibővítése. A Curve osztály lényegében CurvePoint-ok tömbje, ez az osztály reprezentálja az egy-egy játékos által húzott vonalat a játékban.

## Game osztály

A játék modelljét valósítja meg. Tartalmaz egy ServerSidePlayer-ekből álló tömböt a játékos adatok tárolására. Tartalmazza egy mainBoard nevű példányt a Board osztályból. Ez az a tábla, amelyen a játék nyilvántartása történik. Ebben tároljuk az eddig létrehozott vonalak pozícíóit, amikre az ütközés detektáláshoz van szükség. Az updatePosition tagfüggvény fogja végezni a pozíciók aktualizálását, az evaluateStep pedig az ütközés detektálást, és a pontok számítását.

## Board osztály

Tartalmaz mindent, ami a játéktér megjelenítéséhez és a játékosok informáláshoz szükséges, tehát a játékosok által húzott vonalakat. A logikailag inkább a játékhoz tartozó-, de megjeleníteni kívánt adatokat a Game osztályból egy fogadó függvénnyel adjuk át a Board osztálynak, amiket a saját tagváltozóiba ment. A Boardhoz tartozik a játéktér kirajzolását végző függvény, a játéktéren felül a képernyő jobb oldalára kiírja a játékosok neveit és a hozzájuk tartozó pontszámokat.

## PackageS2C osztály

Ez az osztály fogja össze azokat az adatokat, amiket a szerver a kliensnek küld. Ezek a kliens oldalon a játéktér kirajzolásához szükségesek. A játék állapotok szinkronizálásának jegyében az aktuális játék állapotot is megkapja a kliens (ez a játék vége, illetve az előkészület állapotok szempontjából fontos).

## InitPackageS2C osztály

Az inicializációs csomag egy olyan csomag, amit a szerver csak egy alkalommal küld el a kliensnek a játék kezdetekor. A PackageS2C osztályból öröklődik, tartalmazza a játékosok neveit, a hozzájuk tartozó ID-kat (amik egyébként a később tömbként átküldött pontszámoknak a játékos-pontszám összerendelését határozzák meg, az ID a tömbben az adott játékos indexe).

## Játék működése

A játék működését a menü osztály irányítja a legfelsőbb szinten, annak függvényében dönti el a program, hogy éppen a menüt, a játékteret, vagy a játék vége állapotokat rajzolja-e ki a képernyőre.

Szerver üzemmódban a játék először kialakítja a kapcsolatokat a kliensekkel, és elküldi nekik az inicializációs csomagokat. Ezt követően megkezdődik a főciklus, amiben a szerver elvégzi a szükséges számításokat (tehát mozgatja a játékosokat, detektálja és lekezeli az ütközéseket, számítja a pontszámokat), a főciklus utolsó lépése pedig a klienssel történő kommunikáció. A játék futása során jellemzően két állapotban van, az egyik amikor ténylegesen fut a játék és mozognak a játékosok, a másik pedig egy átmeneti, előkészületi állapot a körök között.

Kliens üzemmódban a játék kapcsolódik a szerverhez, majd a kiindulási ponttól kezdve a szervertől minden ütemben megkapja a változásokat, amik alapján frissíti a játékteret. A kliens minden ütemben elküldi a kiválasztott vezérlő billentyűkombinációnak megfelelően azt, hogy a következő ütemben a játékos melyik irányba módosítja a sebességét.