

CLASS DIAGRAMS

I. A pályán elhelyezett objektumok

Konstruktorok

GameObject: A játékmenet során dinamikusan létrejövő objektumok ebből az abstract osztályból származnak. Az objektumok mérete (objectWidth, objectHeight), pozíciója (posX, posY), pozícióváltozása adott idő alatt (deltaPosY, deltaPosX, deltaTime) és az adott objektum sprite-ja (objectAppearance) a leszámazott osztályokban mind megtalálható. Az egyedi attribútumok és a logikát vezérlő metódusok a leszámazottakban kerülnek definiálásra.

GameObjectRenderer: A GameObject leszámazottjainak megjelenítésének megvalósításáért és kirajolásáért felelős osztály.

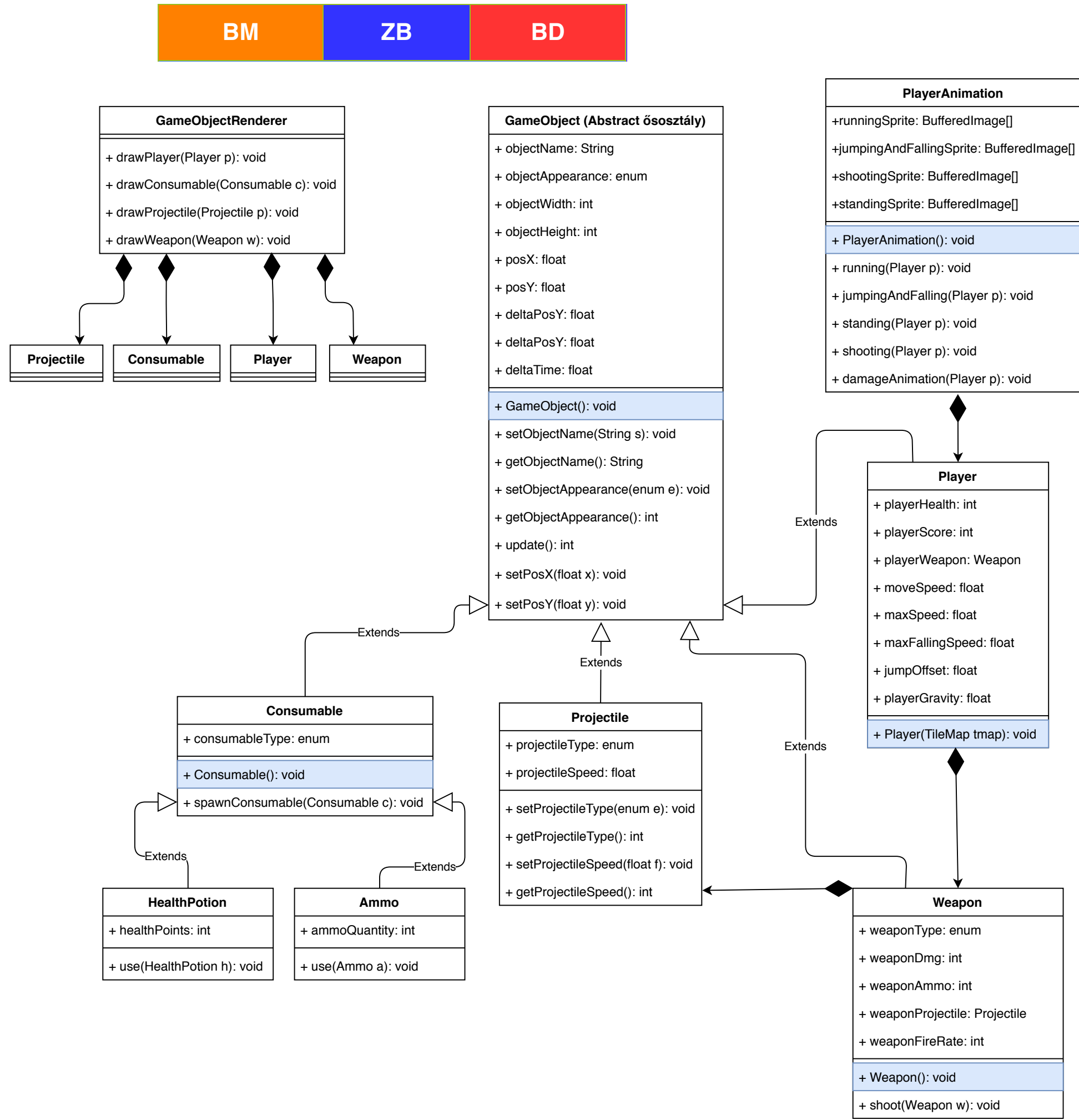
Player: A játékosoknak van életerejük, pontszámuk, fegyverük, mozgási sebességük, esési sebességük, ugrási offsetjük és hat rájuk a gravitáció. A Player egy TileMap-on létezik (ezért szerepel ez a konstruktorban is), ahol a CollisionDetection-t meg kell majd valósítani.

PlayerAnimation: A játékosok sprite kezelése itt történik. Futó, eső / ugró, álló és lövő állapotokat különböztetünk meg. Az adott állapotban egy képsorozatot játszunk le (lehet 1 kép is). Sebészkor a karakter színe vöröses árnyalatot kap.

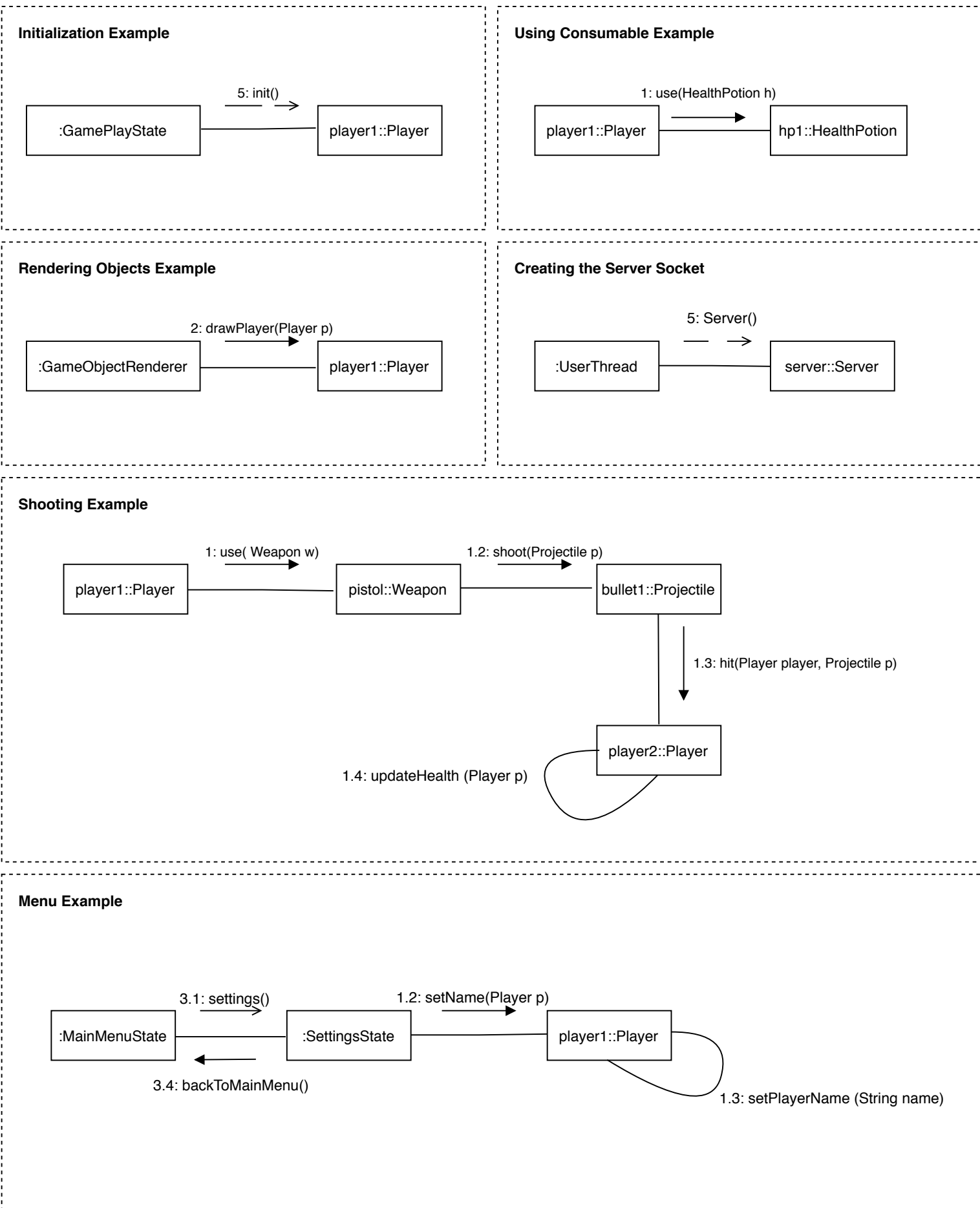
Weapon: A weaponType az egyes fegyvertípusokat különbözteti meg (közvetlenül csak default kerül implementálásra). A Weapon-nak van sebessége, tüzelési sebessége, aktuális töltényszáma. A shoot() metódus indítja el a golyókat, amik ha eltalálnak valamit akkor eltűnnek a képernyőről, különben a mapon megtett távolság függvényében eltűnnek (vagy idő alapján).

Projectile: A lövedékeknek van típusa (fegyvertől függ, de jelenleg csak default típus) és sebessége.

Consumables: A ConsumablesType változóban különböztetjük meg a HealthPickUp-ot és az AmmoPickUp-ot, melyek leszámaznak ebből az osztályból. Eddig a használatkor (felvételekor) healthPoints értékkel növeli a Player aktuális életerejét, utóbbi pedig az ammoQuantity értékkel növeli a Player lőszerének mennyiségét. A "use()" metódusok a Playerrel való érintkezéskor hívódnak meg. (Ezután eltűnnek és a hatásuk végrehajtása megtörténik).



COMMUNICATION DIAGRAMS



II. A rendszer felépítése

Konstruktorok

Start pont

Implementálás miatt definiált metódusok

Main: A játék futtatása innen indul (belépési pont), létrehozuk egy MainProcess példányt.

MainProcess: Az ablak, szálak létrehozása és elindítása, egy StateManager állapotpép példányosítása. A játék gameLoop közben egy képként értelmezhető (BufferedImage), amire ki tudunk rajzolni (Graphics2D) elemeket. A menu Swing alapú, ekkor csak egyszer rajzolunk ki és a gameLoop sem működik. Ezt végzi az updateSwingUI().

updateGame() a logika frissítése, updateScreen() a képen lévő elemek frissítése, majd renderScreen() ezek kirajzolása megadott képrajzolás (FPS szám) szerint.

StateManager: Állapotpép melyben definiáljuk az elérhető állapotokat tartalmazó GameState típusú elemekből álló tömböt, és az aktuális állapotot külön nyilvántartjuk. Az aktuális állapot frissítése – update(), kirajolás – draw(), állapotváltás – setState() és betöltés – loadState(), metódusokat használjuk.

GameState (Abstract): Az osztály konstruktorra egy állapotpépet GameStateManagert hoz létre, ezáltal tudja használni azt. Az abstract metódusokat az egyes State-ekben külön-külön írjuk meg.

MainMenuState: A default (főmenü), mely az indítás után megjelenik. Háttérkép, logo és gombok elhelyezése melyekkel a többi állapot elérhető. Navigálás és kattintás egérrel lehetséges. A gombok színváltása a mouse...() metódusokban minden további állapotban hasonlóan. Állapotváltás az actionPerformed() metódusban gombnyomásra.

JoinState: Ha a helyi hálózaton fut egy szerver, akkor tudunk csatlakozni az éppen futó Sessionhoz és a GameState-be lépünk. A "Back" gombbal visszatérünk a MainMenuState állapotba.

HostState: A "Start" gomb megnyomására az adott gép lesz a Server Host amirez a hálózaton lévő gépek tudnak csatlakozni, GameState-be lépünk. Pályát lehet választani a felsorolt képek kiválasztásával. A "Save" időtartamát pedig egy sliderrel lehet beállítani (1-10 perc). A "Back" gombbal visszatérünk a MainMenuState állapotba.

SettingsState: Az Input mezőben a player nevet adhatjuk meg; a karakterválasztás a maphoz hasonlóan kiválasztható. Az éppen kiválasztott karaktert egy másik kép betöltésével szemléltetjük. A "Save" gomb megnyomásával a változtatások mentésre kerülnek. A "Back" gombbal visszatérünk a MainMenuState állapotba.

ScoreState: A játék végén jelenik meg, ahol lehetőségünk van újrajátszásra "Rematch", visszatérésre a menübe "QuitToMenu", és az eredmények kiírására egy fájlba. Ha már létezik az adott fájl akkor hozzáfűzésekkel kerülnek rögzítésre az adatok, ellenben létrehozuk. A kiírt sor tartalmazza az pontokat és az időpontot is.

HUD: A GameState állapotban megjelenő "statikusan" elhelyezkedő objektumokat tartalmazza a képen (töltény ikon, mellette a töltények száma, aktuális idő, saját és ellenség pontszáma). Ez alól kivétel a karakterek neve és életerejük, mely mindig a mozgó játékosok felett helyezkednek el.

EscapeMenu: Lehetőséget nyújt a kilépésre, azzal hogy az "Esc" billentyűt lenyomva egy panel jelenik meg a képernyő közepén. Az "Esc" ismételt megnyomásával eltűnik az ablak, ha pedig "ENTER"-t nyomunk akkor a főmenübe jutunk.

Tile: Az egyes képek elhelyezése Tile típusú objektumokban.

TileMap: Tile-okat tartalmazó tömb felülírása. txt fájlból. Beolvassuk a sort és oszlopokat majd ezután az adott számoknak megfelelően a griden (mapon) elhelyezzük a megfelelő tile-okat. A draw() végzi az elkészült map kirajolását. A mapon csak statikus tile-ok lesznek. 0 esetén nem rajzolunk.

GameState: A tényleges játékmenet ebben az állapotban történik. TileMap és háttér betöltése, HUD és EscapeMenu objektum létrehozása, Timer indítása, Playerek spawnolása, Consumables (Ammo, Health) és erősebb Weapon spawnolása a pályán, lövedékek létrehozása a shootWeapon() metódusban keresztül.

A játékosok élete csökken ha eltalálják őket és a pályáról leave respawnolndnak (pontlevonással jár, még kérdéses az implementáció). Az életerő folyamatosan frissül, a pontok pedig az okozott sebzés és ölés függvényében kerülnek meghatározásra.

Match leállítása a matchDuration eltelte után. Ekkor megjelenik a ScoreState és az ott megjelenő opciók közül választhatunk.

