



A program felépítése a következő:

-A main file-ban megnyitok egy ablakot Raylibbel, utána létrehozok egy World-öt és amíg nyitva van az ablak, meghívom az Update és Draw metódusait.

-A világ tartalmazza a játékost, az összes Coin-t, Enemy-t és Bullet-ot.

-A játékos figyeli a billentyűzetet és az egeret a mozgáshoz, az Enemy-k pedig automatikusan mozognak a játékos felé. A Coin-ok és a Bullet-ok szintén maguk mozognak.

-Minden objektum, többek között azok is, amiket felsoroltam a GameObject osztályból származik. A GameObject osztály pedig tartalmaz egy World\*-t, ami az őt tartalmazó osztályra mutat. Ez azért szükséges, mert ha a játékos lő, akkor létrehoz egy Bullet-ot, és hozzá kell adnia a World-höz, amiben van. Lehet, hogy lenne ennél jobb megoldás is, de azért nincs vészesen sok objektum, szóval egy pointer talán nem nagy gond (viszont nagyon megkönnyíti a dolgom).

-A World.Update update-el mindent (mozgat), ezen felül ellenőrzi a következő ütközéseket: játékos - Collectable, játékos - Bullet (jelenleg ilyen nincs), Enemy - Bullet, játékos - Enemy. Ha ütközés történt, törli a megfelelő objektumokat.

-A ParticleSystem egyelőre egy helyen jelenik meg, méghozzá a játékos hoz létre egy példányt, és amikor mozog, meghívja az Emit metódust, ami kibocsát néhány Particle-t. Természetesen a játékos Update metódusa meghívja a ParticleSystem Update metódusát, az pedig a Particle-ök Update metódusait, így idő után a Particle-ök eltűnnek (a ParticleSystem törli a öreg Particle-öket).

-A különböző méretű ellenfelek valóban csak a méretükben és a sebességükben térnek el (de egy kicsit máshogy is vannak rajzolva), így tényleg kicsit felesleges az öröklődés, de eredetileg terveztem ennél nagyobb eltérést is (egyelőre nincs).

-Ahogy gondoltam, a rajzolás során néhány képet is kirajzolok, a teszteléshez kivettem a grafikus részeket, de azon kívül nagyjából mindenhez írtam egy rövid tesztet. Sajnos a Raylib függőséget csak úgy sikerült kiküszöbölnöm, hogy a rajzoló részeket meg a többi Raylib-et használó részeket egy makróval kikapcsoltam (nem tudom ez mennyire jó megoldás, de legalább működik).

-Igen, használhattam volna a Vec2 helyett egy kész könyvtárat, lehet jobb is lett volna, de valahol szerettem volna használni template-et és itt kézenfekvőnek tűnt.

-A teszteléshez a CMakeLists.txt file-ban egy sort kell csak átírni a tetején (set(CPORTA on))