Tutorial – rpg test

1. **NPC és Player renderelési sorrendje:**Mind a Player és NPC objektumok egy Circle Collider 2D-t és egy Box Collider 2D-t tartalmaznak. A Circle Collider 2D az objektumok egymáshoz ütközéséért és egymáson való átgördüléséért felelős. A Box Collider 2D pedig Trigger-ként működik, hogy érzékelni tudjuk amikor két Sprite egymáson helyezkedik el és el kell döntenünk, melyik legyen felül. A hero.cs-ben kezeljük le az OnTriggerEnter2D eljárás automatikus meghívásával, ha a Player-ünk Trigger-ével másik Trigger érintkezik (találkoznak). Amennyiben az NPC Trigger-ét érzékeljük, eldöntjük hogy az NPC vagy a Player helyezkedik-e el magasabban (távolabb) a síkon és ennek megfelelően beállítjuk a Player rendezési tényezőjét. Az NPC-nek 1-et állítottunk alapértelmezett rendezésnek, tehát a Player a 0 vagy a 2 értékeket veszi fel. Ha a sorrend megváltozna, pedig a Trigger-ek még mindig folyamatosan metszik egymást akkor az OnTriggerStay2D hívódik meg, ahol az előbb leírt eljárást követve újra meghatározzuk a sorrendet és változtatunk rajta, ha szükséges.
2. **Player és háttérbeli objektumok (korlát, pad) renderelési sorrendje:**Az objektumokat a háttérből kivágva 1-es sorrenddel vittem a jelenetbe. Az 1. pont mintájára Trigger-eket helyeztem el OutSide és InnnerSide néven. A velük való érintkezést szintén a hero.cs fájl OnTriggerEnter2D eljárásában érzékeljük, ahol OutSide esetén a Player-t az objektum mögé (0-s rendezés), míg InnnerSide esetén az objektum elé (2-es rendezés) tesszük.
3. **NPC helyes irányba nézése:**Az Npc.cs fájlban a FixedUpdate eljárásban tükrözöm az NPC-t annak megfelelően, hogy az útvonal következő csomópontja az előzősomóponttól jobbra illetve balra helyezkedik-e el.

Az említett kódrészleteket a forráskódokban kommentekkel is jelöltem.