

# Szakdolgozat I. Beszámoló

**Cím:**

**Készítette:** Demján Csongor

**Neptun:** SZ7MGG

**Témavazető:** Dr. Nagy Noémi

## Bevezető

Egy HTML5 2D oldalnézetes platformer játék tervezésére, majd fejlesztésére törekedtem ebben a félévben, melyben a karakter idősíkok közötti folyamatos ugrálással/változtatással tud tovább haladni. Vanilla JavaScript, azaz az eddigi kódolás során külső library-t és framework-öt nem használtam. A fejlesztés folyamata alatt ez idáig 13 lépést sikerült teljesítenem, melyek amolyan mérföldkövek voltak a mostani állapot elkészítéséhez.

## Feladatok és megvalósítások

A legelső feladatom a játék alapkoncepciójának kitalálása volt. A tervezési fázisban irányt mutatott, hogyan lehetne az idősíkok közötti váltakozást mutatni. Így megszületett egy olyan játék, amelyben a karakterünk 2 háttér, és az ezekhez rendelt tulajdonságok között tud váltani. Még a játék élvezhetőségi szintje nem állja meg a helyét, viszont úgy gondolom, hogy minden olyan alapelem, amely szükséges a játék alapjául sikeresen elkészült. Jelen pillanatban egy olyan pályát tervezek, amelyen a karakter a szükséges képernyőcserét szükséges használni, hogy tovább haladjon a következő szintre.

A tervezést követően az első lépésem a projekt elkezdéséhez szükséges alapbeállításokból állt. Létrehoztam egy meghatározott nagyságú hátteret, majd ezen kirajzoltattam egy eltérő színű négyszöget, létrehoztam neki a player osztályt. A következő lépésben a gravitációt adtam hozzá, itt deklaráltam egy változót, amely a gravitációs gyorsulást jelképezi a canvason. Itt gyakorlatilag az történt, hogy a kirajzolt négyzet helyzete folyamatosan frissült az y tengelyen, mivel realisztikus hatást szerettem volna, így létrehoztam egy gyorsaság változót, amelyet, ha folyamatosan hozzáadunk a gravitációhoz egy gyorsulást eredményez. Ez a változó kapott egy x értéket is az y értéken kívül, amely az oldal irányba való helyzetváltoztatáshoz lesz majd szükség. Alapból kiesett a képernyő alján, viszont ezt gyorsan lehetett javítani, meg kellett határozni, hogy a canvast nem hagyhatja el az alján. A négyzet mozgásához a gyorsulás x változóját használtam, miután hozzáadtam két eventListanert, ahol a mozgáshoz szükséges lenyomott billentyűk lenyomását és felengedését figyeljük. Például a D billentyű lenyomásakor folyamatosan hozzáadunk a gyorsulás x változójához, a W lenyomásakor az y változóhoz. Felengedéskor biztosra kellett menni, hogy nem tartja meg a gyorsaságát a karakter, így ezt ebben az esetben visszaállítottuk 0-ra. A háttér sprite létrehozásakor létrehoztam a Sprite osztályt, ahol megadtam a pozíciót és a képet, majd a hátteret le kellett kicsinyíteni, mivel nem szerettem volna, ha a nagy háttér egyben látszik. Ezt az követte, hogy meghatároztam a kép betöltésekor melyik része látszódjon. A háttérhez rendelt collision block-ok elkészítésében

segítséget nyújtott a Tiled mapeditor, amelyből tudtam .js kiterjesztésben exportálni. Ebből kimásoltam a szükséges adatokat és létrehoztam egy külön fájlt ezek eltárolására. Létrehoztam a szükséges metódusokat, amelyek képesek ezeket az adatokat kiolvasni, és olyan értékkel felruházni, amely lehetővé teszi, hogy a képernyőn kirajzoljuk ezeket a blockok-at, úgy, hogy a karakter alja érzékelje amikor hozzáér egy block-hoz, és a tetején maradjon. Amikor a collision detektálást készítettem egy kicsit el kellett játszani azzal, hogy azt az eredményt érzük el, ami élvezhetővé körülményeket ad a játékhoz. Ezeket a lépéseket 2 részre bontva sikerült megvalósítani, még hozzá úgy, hogy az alapot és a levegőben lévő blockok-at külön szedtem és mindkettőn alkalmazzuk a collision detection-t, csak egy kicsit eltérő módon, hiszen a levegőben lévő blockok-ra úgy szerettem volna tudni felugrani, hogy fel lehessen ugrani rájuk. Mivel ekkor már volt egy háttérünk, úgy döntöttem ideje a karakternek is adni egy kinézetet, mind a háttér és karaktert egy ingyenesen letölthető asset-ként találtam, és használtam fel a játék implementációjában. Hozzáadtam a szükséges .png fájlokat a projekthez, majd elkezdtem implementálni a sprite animációt és ezeknek a cseréjét. Kiegészítettem a Sprite class, hogy a karakter képének betöltésekor ezt a képet rajzolja ki, majd, hogy ez a kép folyamatosan változzon annak megfelelően, hogy milyen állapotok vannak megrajzolva. Hozzá kellett rendelni a különböző képeket, amelyek a karakter mozgásának állapotaira vannak bontva az egyes mozgási állapotokhoz. Azt az eredményt várva, hogy amikor ugrik a karakter, akkor az ehhez illeszkedő képet használjuk. Ezeket a képeket a különböző irányok is meghatározták, ezért amikor a karakter bal irányba néz, mozog, ugrik, esik az ezen irányhoz tartozó kép kerüljön betöltésre. Trükkös volt megválasztani milyen gyorsan változzon a kép az animáció során úgy, hogy realisztikusabb és szebb legyen az animáció. A hitbox implementációja következett. Ez azért volt szükséges, mivel amikor betöltjük a karakter képét, körülötte van egy olyan négyzet, ami meghatározza a collision detection-t, és szerettem volna elérni azt a hatást, hogy a karakter csupán a közvetlen körülötte lévő helyzet által meghatározottan tudjon mozogni. Implementálni kellett egy konténert, amely körbe határolja a karaktert, és felhasználásával a karakter közvetlen a talajon tudjon állni, és kibővíüljön az x tengelyen való mozgástartomány. Ezt követően meghatároztam, hogy a box ne hagyja el képet a canvas szélein. Horizontális kamera követést szerettem volna elérni, azaz létrehozni egy olyan négyzetet a karakter köré, amely nagyobb annál, viszont kisebb azon képnél, ami látszik a játék háttérének megjelenítésekor. Amikor jobb oldali irányba megyünk a játékosunkkal, szerettem volna elérni azt a hatást, hogy a kép háttere eltolódjon abban az esetben amikor ez az előbb definiált négyzet hozzáér a canvas széléhez, és a mozgásnak megfelelően kövesse a háttér a karakter helyzetét. Viszont meg kellett határozni a végeket, azaz, hogy a kép ne tudjon túltolódni a háttér képének határain és ekkor egy üres képet jelenítsen meg. Ezt követően a követően azt kellett elérni, hogy a karakter tudjon az egész háttérén mozogni, tehát meg kellett határozni, hogy amikor a szélekhez ér, akkor ne tolja tovább a háttér, viszont az eltoláshoz használt háttér ne befolyásolja a karaktert a map széléhez való eljutásban. Miután ezzel megvoltam, hasonló elveket felhasználva elkészítettem a vertikális kamerakövetést. Így amikor a karakter az y tengelyen mozog a kamera szintén követni fogja a helyzetét és eltolja a háttér.

## **Az idősíkok közötti különbségtétel**

Mint ahogy említettem a bevezetőben ezt a hatást úgy értem el, hogy 2 hátteret az adott szinthez rendelve cserélni tudunk a hozzájuk rendelt tulajdonságokkal együtt. Beállítottam, hogy az 'L' billentyű lenyomásakor a létrehozott maps változó 1-es és 2-es kép és szükséges tulajdonságai között tudjon változtatni, ehhez szükséges volt a map változó deklarálása, mely alpból 1, mivel az első képet szeretnénk látni az induláskor, viszont lenyomáskor változz 1-ről 2-re, és amikor ebben a fázisban van 2-ről 1-re.

A kód további implementációs tervet tartalmaz.