

A mozgás és a játékélmény összekapcsolása

Szücs Barna

Szücs Barna

Felkészítő tanár: Vereb Márta

BGSZC Mechatronikai Technikum, 1118 Budapest, Rétköz utca 39.

1. Bevezetés

A mai társadalomban gyakori a játékfüggőség és a részben ehhez köthető elhízás. Az elhízásra egyszerű megoldás lenne a testmozgás, de sok ember nem hajlandó ennek eleget tenni részben a játékfüggőség következményeképpen. A célom ebben a projektben egy egyszerű Python program elkészítése, ami egy élvezhető játékban testesül meg és mozgásra készíti az embereket.

2. Probléma megoldásának menete

Játékként a Flappie bird-öt választottam, ami egy régi és egyszerűen elkészíthető játék. Egy madárról szól, akinek repülés közben csöveket kell kikerülnie. Ezek a csövek 2-es párokban egymás fölött helyezkednek el, egy kis (megadott szélességű) rést hagyva maguk között a madárnak. A játék egy jelzéssel irányítható, ami az ugrásra készíti a madarat. Amennyiben a madár földet ér, a játék véget ér. Ha a madár csőbe ütközik, ugrásképtelenné válik, lezuhan és így módon ér véget a játék. A madár nem képes a képernyőből kimenni (a plafonról visszapattan).

Ami mozgásra készíti a játékost, az a játék irányítása --> az ugrás jelzése a végtagok/test mozgatásával lehetséges. Ehhez a számítógép kameráját használok.

2.1. A programkód

A programkód két kisebb program egybekötésével készült.

2.2. A "Flappie bird"-nevű játék általam programozott (szerény grafikájú) verziója

A megjelenítéshez turtle-t használok, a madár x-koordinátája rögzítve van a képernyő közepéhez és csak az y-tengely mentén mozoghat (zuhanás/ugrás). A csövek közötti rés y-tengely menti elhelyezése randomizálva van, de a szélessége megadott hosszúságú. A framek egy while True-ciklussal vannak megoldva, ahol elsőként a program frissíti a képernyőképet, a madár y-tengely menti sebességét csökkenti (gravitációs effekt), azután pedig az új sebességeknek megfelelően mozgatja a madarat és a csöveket (a csövek x-tengely menti sebessége adott). Miután a képernyőn lévő "szereplőket" elmozgatta, a program ellenőrzi a tárgyak új pozícióit:

- ha a madár belemegy a plafonba akkor az y-tengely menti sebességét lenullázza

- ha madár földet ér akkor a játéknak vége és a program újra indítja önmagát, hogy új menetet kezdhesen
- ha a madár áthaladt egy résen, akkor a pontszámláló pontszámához hozzáad 1-et. Ez az eset úgy van megállapítva, hogyha az (1)-es és (2)-es egyenlet egyaránt teljesül.

$$MadárXKoord - CsőXKoord > \frac{(MadárSzélesség + CsőSzélesség)}{2} \quad (1)$$

$$MadárXKoord + \frac{MadárSzél - CsőSzél}{2} - CsőXKoord \geq CsőXSebesség \quad (2)$$

Ahol

- MadárXKoord – A madár X koordinátája
 - CsőXKoord – A cső X koordinátája
 - MadárSzél (vagy MadárSzélesség) – a madár X tengely menti kiterjedése pixelekből kifejezve
 - CsőSzél (vagy CsőSzélesség) – a cső X tengely menti kiterjedése pixelekből kifejezve
 - CsőXSebesség – A cső x tengely menti sebessége
- ha a madár csőbe ütközik akkor nem engedi, hogy ugorjon és a játék vége előtt egy zuhanás jelenetet hoz létre. Ez az eset akkor áll fenn, ha a (3)-as és (4)-es egyenlet egyaránt teljesül.

$$|MadárXKoord - CsőXKoord| < \frac{(MadárSzélesség + CsőSzélesség)}{2} \quad (3)$$

$$|MadárYKoord - CsőYKoord| < \frac{(MadárSzél + CsőHossz)}{2} \quad (4)$$

Ahol CsőHossz – a cső Y tengely menti kiterjedése pixelekből kifejezve.

- ha az egyik csőpár egy cső fentről a másik lentől jön) kimegy a képernyőből akkor egy másik csőpár mögé helyezi új y-tengely menti magassággal (rés magasságát változtatja)

A frame futtatása közben a program méri az időt is és a frame végén annyit vár amennyi még hiányzik az előre megadott frame-időhöz.

2.3. Egy program, ami a kamera aktuális képéből megállapítja néhány a testen elhelyezkedő pontnak a koordinátáit

A pontok megtalálásához a Google Mediapipe programcsomagját használom. Az ugrás-jel küldéséhez sok feltételt megadhatunk (pl. ha a könyök csúcsa magasabban van, mint a váll, akkor ugrás-jelet küld). Ebben a megoldásban a jel küldésének feltétele az, hogy a könyök sebessége nagyobb legyen egy

megadott értéknél. A megoldásban ezt a feltételt használtam, de kiemelném, hogy egyéb pontokat és azok kapcsolatát is ugyanilyen módon lehet elemezni és egy intenzívebb mozgást megadni feltétel-ként. Számomra a legérdekesebb a sebesség kiszámítása volt. A frame-enkénti sebesség kiszámolásához két frame-ben meghatározott koordinátákra van szükség és a frame-ek futási idejére. A számítási képlet a következő:

$$\frac{\sqrt{((Frame1Y-Frame2Y)^2+(Frame1X-Frame2X)^2)}}{Frame1Idő} \quad (5)$$

Ahol

- Frame1Y – az adott pont első frame-beli Y koordinátája
- Frame2Y – az adott pont második frame-beli Y koordinátája
- Frame1X – az adott pont első frame-beli X koordinátája
- Frame2X – az adott pont második frame-beli X koordinátája
- Frame1Idő – az első frame lefutásának ideje

Mivel a framek ideje meg van szabva ezért csakis akkor lehet eltérés, ha egy frame a megadott határértéknél lassabban fut le, de ezt a program figyelembe veszi. Természetesen a programban nem csak az utolsó 2 frame-ben eltelt változásokat figyeljük, ugyanis ily módon pontatlanságok fordulhatnak elő. Ezen pontatlanságok elkerülése érdekében az utolsó 5 frame-ben megállapított sebességet raktározzuk el és azoknak az átlagát vesszük figyelembe a feltételnél.



1. ábra: A Mediapipe-pal azonosítható pontok elhelyezkedése a testen és kamerakép, rajta a megállapított pontokkal

3. Elért eredmények

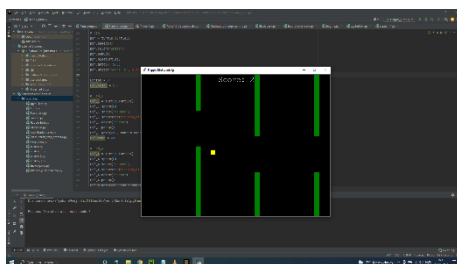
A flappie bird program megvalósítása sikeres volt, bár a madár csak egy sárga kockaként van feltüntetve, de ez nem probléma, hiszen ez a projekt csak egy példa a testmozgással való játék irányításra.

A képelemző program is sikeres lett, pár tized másodpercnyi késéssel megállapítja a sebességet.

A kettő összekapcsolása is teljesült, és bár kicsit lassú a program, de ez nem okoz akadályt a játékban és még mindig élvezhető. A flappie bird játék

újraindítás funkciója nem lett kifejlesztve a projekt során, ezért minden menet után újra kell indítani a programot manuálisan a következő menethez.

Összességében elértem az eredeti célomat: már pár percnyi játék során is komoly fizikai mozgást igényel a játék irányítása. Továbbfejlesztési lehetőségnek az egyéb mozgásformákkal való irányítási módok kidolgozását látom. Ezáltal egy a teljes testet megmozgató, de élvezetes játékot fogunk létrehozni



2. ábra