

# Fejlesztői dokumentáció

Vincze Csongor

December 10, 2025

## Contents

1 A program ismertetése, áttekintése	3
2 Nyelv, használt csomagok, kiegészítők	3
3 Az arhitektura áttekintése, a program kezelése	3
4 Fájlok és azok leírásai	3
4.1 vec_3.h . . . . .	4
4.2 ray.h . . . . .	4
4.3 sphere.h . . . . .	4

## 1 A program ismertetése, áttekintése

Ez a program egy nagyon alapvető sugárkövetőt (ray tracer) implementál le. A lényege, a virtuális kamerából (ahonnét mi a képet látjuk) sugarakat küld, és ezeknek a sugaraknak a gömbökkel való ütközését követi. Miután a felhasználó megadja a megfelelő paramétereket a program elkezdi a képkockák generálását. A képkockák generálásának előrehaladását közben a felhasználó egy folyamatjelző sávon követheti. Amint az összes képkocka generálása megtörtént a szoftver összefűzi a képkockákat egy videová, és ezt el is indítja.

## 2 Nyelv, használt csomagok, kiegészítők

A teljes projekt C nyelvben íródott, az alapvető C könyvtárak felhasználásával, minden külső csomag nélkül. A program nagy vonalakban a <https://raytracing.github.io/books/RayTracingInOneWeekend.html#overview> dokumentációt követi (természetesen C nyelvre átírva).

## 3 Az arhitektura áttekintése, a program kezelése

A `video_render.exe` futtatásakor a program bekéri a videó elkészítéséhez szükséges adatokat. Ezek sorba:

1. Milyen felbontást szeretnél? (szélesség pixelekben) (elfogadható értékek:  $2^n$ , ahol  $n$  egész és  $4 < n < 11$ )
2. Hány gömböt szeretnél? (ajánlott: 2-10)
3. Milyen hosszú videót szeretnél? (a videó 30fps-en fut, 1-10 közötti egész szám elfogadott)

Az első bemenet a kép vízszíntes pixelszámát adja meg (a képarány 16:9). Itt 1024-nél már jelentősen lelassul a program. 256-nál viszont még elég alacsony lesz a készített videó minősége. Érdemes lehet ezt az értéket 512-re állítani. Ezzel az adattal a program már el tudja készíteni az ún. nézőportált (viewport), ami meghatározza, hogy mit, hogy fogunk látni. A többi paraméter fix. A `viewport_creator` függvény végzi ezt a folyamatot, és a `video_render.c` hívja meg a `render.h`-ból. A gömbök száma egészen 0-tól 50-ig terjedhet. Ennek ellenére nem ajánlott 10-nél többet megadni mivel ez is jelentősen lelassítja a programot. A gömbökre van egy külön struktúra (sphere), egy ilyen tömböt hozunk létre. A a videó készítés közben mozgatni fogjuk a gömböket. Mivel csak gömb alakú objektumokat dolgozunk, így érdemes ennek a tömbnek az elejére egy előre beállított nagy gömböt rakni, hogy legyen egy földünk a videóhoz. Így valójában egyel több gömbbel operálunk mintamennyt bekértünk. A harmadik bekérésnél hasonló okokból kiindulva 5 körüli értéket érdemes megadni. Mivel konstans 30 fps-el dolgozunk így csak simán visszaszámoljuk, hogy hány képkockát kell legenerálunk.

## 4 Fájlok és azok leírásaik

A program egy fő .c fájlból és 10 header fájlból áll. A header fájlok főleg a program egyes részeihez biztosítanak függvényeket. minden fájl ugyan abba n a mappában van.

## **4.1 vec\_3.h**

Mivel 3 dimenzióban dolgozunk szükségünk van vektoroperációkra. Ez a fájl biztosítja ezek meglétét. A fájlból minden függvény mellé fel van tüntetve, hogy mit csinál, így ezt nem részletezem itt. Ez a fájl tartalmazza a `vec_3.h` struktúrát is ami, lényegében egy 3 dimenziós vektor típus.

## **4.2 ray.h**

Ez a rész csupán a sugár (ray) struktúrát és egz sugárkövetőt tartalmazza. A sugár egy kiinduló pontból és egy irányvektorból áll. A sugárkövető pedig visszaadja egy t paraméter függvényében, hogy hol van a sugár.

## **4.3 sphere.h**

Ez a fájl a gömb struktúrát hozza létre. A gömb egy középpontból és egz sugárból áll. Ezután a `hit_sphere` függvény következik ami, lényegileg eldönti, hogy az adott sugár talált-e és ha igen melyik gömböt találta el. Ezt két függvény követi; az `_array_spheres` és a `_rand_spheres`. Előbbi egy vízszintes sorba rak le gömböket. Útóbbi pedig (bizonyos korlátok kötött) random paraméterekkel rakja le a gömböket.