

The Backrooms

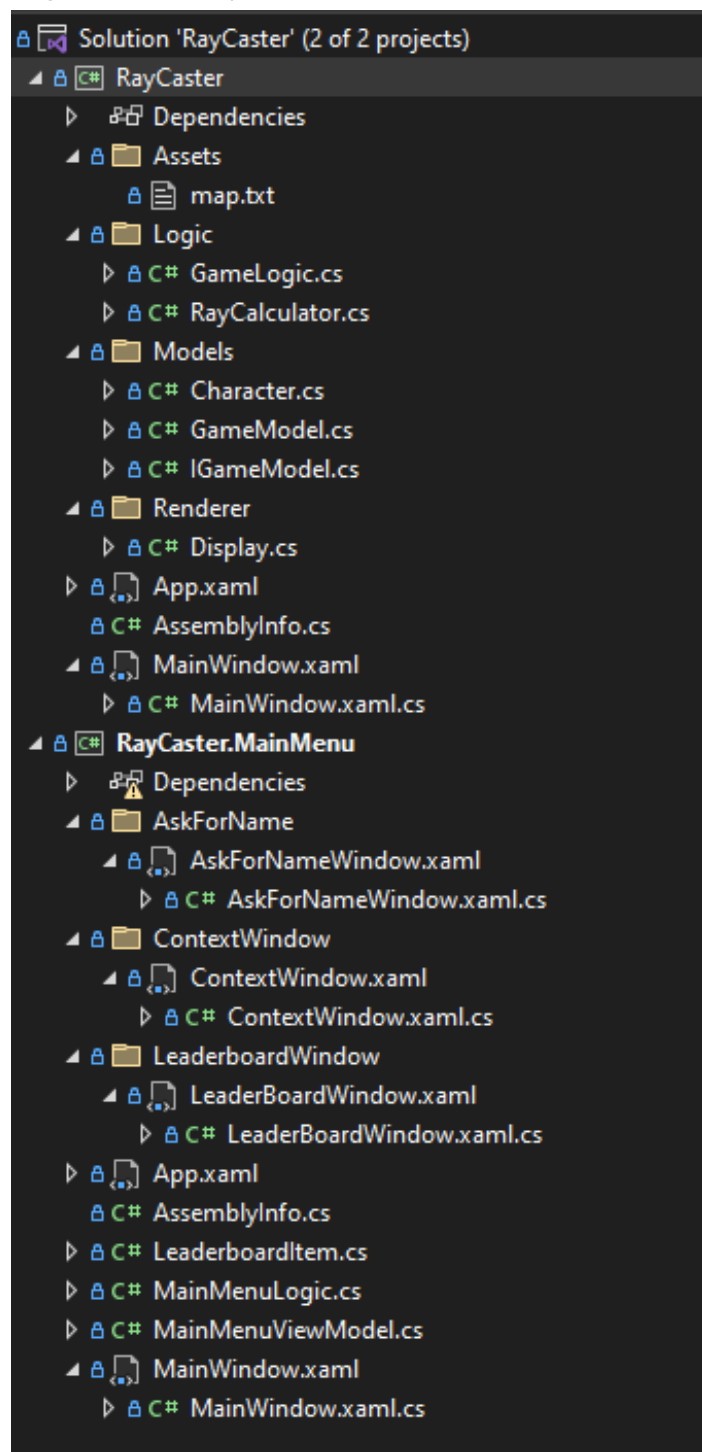
A játékról röviden

A játék egy raycaster alapú labirintus játék, ami a “backrooms” városi legendát kelti életre. A “backrooms” egy olyan hely, ahová – a legenda szerint – a valóságból lehet bejutni, ha nem vigyáz az ember. Ebben az irodahelyiségben örökké bókászhat bárki, aki bekerül, soha nem fog kiutat találni.

Referencia:



A játék felépítése



A program tulajdonképpen két fontosabb részből áll, az egyik maga a játék, a másik, a játékhoz tartozó menürendszer.

RayCaster:

Logic: Itt található a játékhoz tartozó irányítás, illetve a sugarak, amikkel a falakat ábrázoló téglalapok magassága kiszámolható.

Models: A karakter és a pálya tulajdonságai.

Renderer: Itt található a Display, ami a tényleges kirajzolást fogja végezni attól függően, hogy a **RayCalculator** milyen adatokkal tér vissza. Ha térkép módban vagyunk, akkor pedig a térképet felülnézetből rajzolja ki.

MainMenu: Ebben a projektben az összes ablak mvvm vagy szimpla C# WPF alapokra épült, komplexitástól függően.

MainWindow: Ez az ablak fog elindulni a játék indításakor, innen tudunk a tényleges játékba bekerülni, a leaderboard-ot megnézni, olvasni a városi legendáról vagy kilépni.

AskForName: A játék indításakor meg kell adnunk egy nevet, amely a leaderboard-on szerepelni fog.

ContextWindow: Itt

olvashatunk egy kicsit a városi legendáról.

LeaderboardWindow: Nevek, a játékban töltött idejünkkel párosítva.

RayCasting

A raycasting technológiát fogja a játék alkalmazni. Ezt a rendszert használta a régi Wolfenstein 3D, Doom és egyéb klasszikusok is.

Az alapelv az, hogy van egy 2D-s térképünk, amelyekben '0'-k jelzik a folyosót, és '1'-k jelzik a falakat. (Később lesz szó róla a *bővítés* szekcióban.)

A játékosnak van egy FOV-ja (field of view/látószög), amelyen belül ki fogunk lövelni sugarakat. Ha ezek a sugarak fallal ütköznek, akkor kiszámítható, hogy egy falnak mennyire kell magasnak lennie, ezzel térhatást keltve. Továbbá a falakról azt is meg lehet mondani, hogy melyik oldalát látjuk, a játékostól oldalra eső falakat kissé sötétebb színnel fogjuk megjeleníteni.

A padló és a plafon egyszerűen megvalósítható, hiszen minden fal a képernyő közepétől fog kiindulni. Ezt kihasználva, még a falak kirajzolása előtt, feltöltjük a képernyőt a középvonaltól felfelé a plafon színével, lefelé pedig a padlóéval.

Bővítés

Ha a projekt gyorsan halad, az bővíthető egyéb színű falakkal, hiszen a RayCalculator bővíthető úgy, hogy a falnak a színét is visszaadja. Például a térképen '2' jelölné az egyéb színű falakat. Képeket is lehet akár használni, nem csak színeket, de ez meglehetősen nehezebb, hiszen a képeket pixeloszloponként kell felépíteni, ahhoz, hogy szépen nézzen ki a játék.

Fejlesztés lépései

Timer

A logika nagyrésze azon alapszik, hogy másodpercenként x alkalommal meghívódnak a metódusai. A kirajzolás (és a hozzá illő számolás), illetve a mozgás is másodpercenként 60-szor lesz meghívva, ez természetesen állítható, kisebb teljesítményű rendszerek esetén. Ehhez a Timer-nek ~16ms-enként kell ütnie.

Kirajzolás

Első lépésként véletlen színű 1 pixel széles oszlopokat rajzolunk ki a képernyőre. Innentől kezdve ezeknek a mérete a később megírandó RayCalculator alapján kiszámítható.

RayCalculator

Itt fog a legtöbb számítás lezajlani. Ennek az osztálynak a logikája alapján kiszámítható lesz a falak magassága a Display osztály számára, és máris ki tudunk rajzolni egy pályát „3D”-ben.

Mozgás

El kell érni, hogy tudjunk mozogni a játékban. A Keyboard.IsKeyDown osztályt használva egyszerre akár több gombot lenyomva is mozoghat a játékos. Tehát ha például W és A gombot is lenyom, akkor balra és felfelé is mozogni fog egyszerre.

Hitbox

A játékos már tud mozogni, azonban a falakon még át tud sétálni. Ez nem csak logikátlan, hanem a játék összeomlásához is vezethet, hiszen a raycaster rendszer azon alapszik, hogy mindenképpen lesz mivel „ütköznie” a sugaraknak.

Könnyen megoldható, hogy a játékos ne tudjon áthaladni a falakon, szimplán mozgás előtt elmentjük a helyét a játékosnak, és ha a lefelé kerekített pozíciója egy falon belül esne, akkor a mozgást „visszavonjuk” (mivel minden a rajzolás előtt lezajlik, ezért nem lesz villódzás, ebből a játékos semmit sem érzékel).

Térkép mód

Írjunk meg egy térképes üzemmódot, amire az 'M' betűvel válthatunk. Itt kirajzoljuk felülnézetes módban a játékkeret, a karakterünket és szimuláljuk a sugarakat, amiket kilövünk.

Menürendszer

Játék indítása

Tudjuk elindítani a játékot, egyszerűen megnyitjuk a játék ablakát, ami innentől az előzőek szerint működik.

Leadeboard

Mivel a játék eléggé meta, és nincs tényleges célja, ezért abban tudjuk mérni a játékosok teljesítményét, hogy mennyi időt sikerült elpazarolniuk a backrooms-ban. Ehhez a játék indításakor bekérünk egy nevet, majd az indításkor lementett idő, és a befejezéskor lekért idő alapján ki tudjuk számítani, hogy mennyi időt töltött a játékos a labirintusban.

Context

Mivel egy városi legendáról van szó, nem mindenki fogja ismerni. Ezért létrehozunk egy ablakot, amiben röviden leírjuk azt, amit a dokumentum elején is leírtunk.

Kilépés

Természetesen a játékot be is kell tudni zárni. A játék alapvetően nem fogja a felső „X” gombot és társait tartalmazni, így csak a kilépés gombbal tudjuk elhagyni a játékot.