The Backrooms

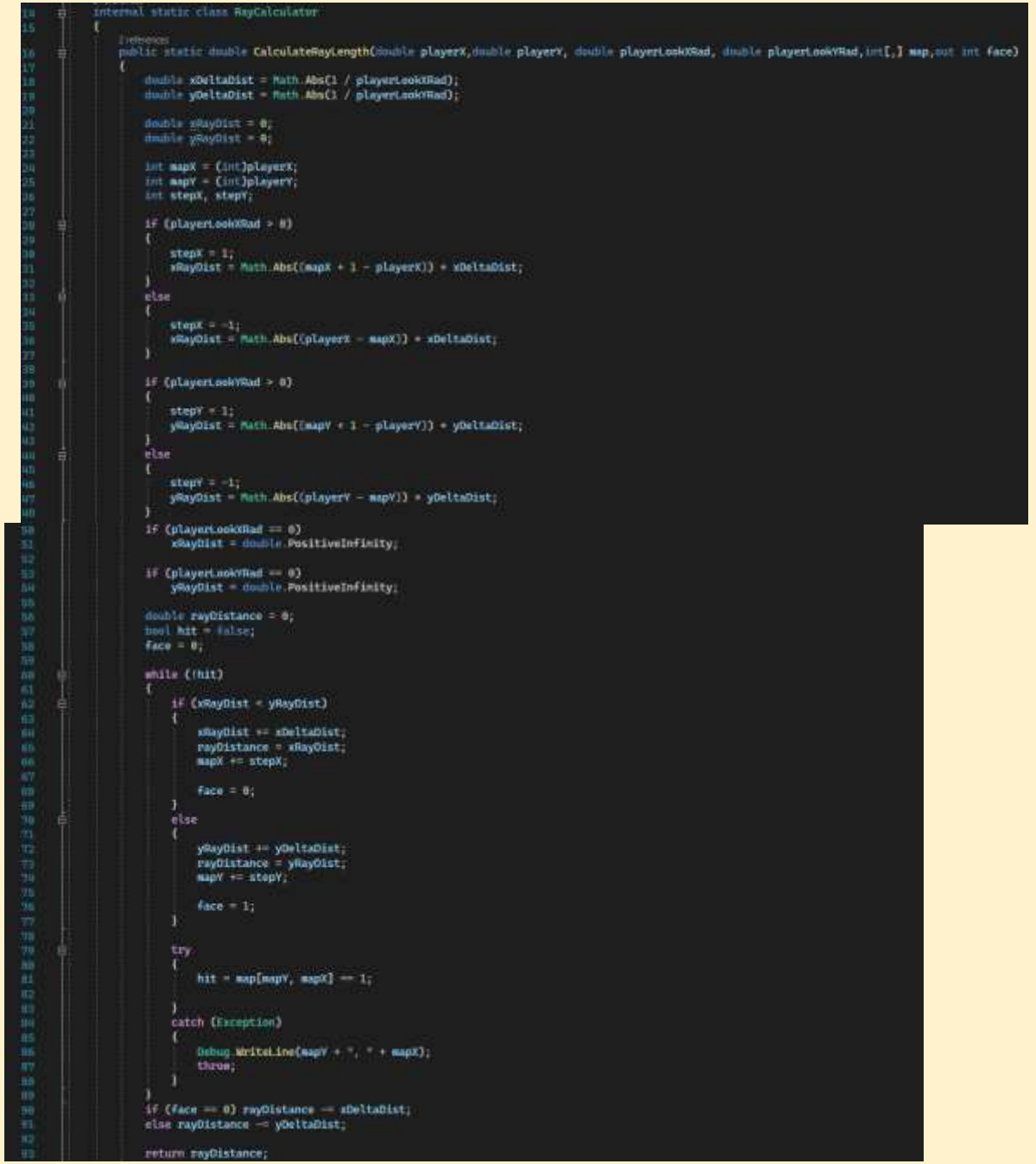
Készítette: Kvaka Martin – 12 commit – 4-es jegy

Kis kontextus

- A Backrooms egy városi legenda, miszerint, ha nem vagyunk ügyesek, ki lehet “noclip”-elni a valóságból.
- Ekkor egy ehhez hasonló helyre kerülünk, és ott örökké kereshetjük a kiutat, ami nem létezik.
- Ez alapján csináltam ezt a játékot, aminek “semmi értelme”, nagyon meta…
- A tantárgyhoz a magyarázat inkább csak annyi, hogy egy raycastert szerettem volna írni, úgy ítéltem ez egyedül menni fog.

Legnehezebb feladat

Habár a RayCalculator osztályt egy C++ guide segítségével írtam meg, ettől függetleül ez volt a legnehezebb része a projektnek. Egyrészt megérteni a matekot, ami a kód mögött áll; másrészt “átfordítani” C#-ra.



```
internal static class RayCalculator
{
    1 reference
    public static double CalculateRayLength(double playerX,double playerY, double playerLookXRad, double playerLookYRad,int[,] map,out int face)
    {
        double xDeltaDist = Math.Abs(1 / playerLookXRad);
        double yDeltaDist = Math.Abs(1 / playerLookYRad);

        double xRayDist = 0;
        double yRayDist = 0;

        int mapX = (int)playerX;
        int mapY = (int)playerY;
        int stepX, stepY;

        if (playerLookXRad > 0)
        {
            stepX = 1;
            xRayDist = Math.Abs((mapX + 1 - playerX)) * xDeltaDist;
        }
        else
        {
            stepX = -1;
            xRayDist = Math.Abs((playerX - mapX)) * xDeltaDist;
        }

        if (playerLookYRad > 0)
        {
            stepY = 1;
            yRayDist = Math.Abs((mapY + 1 - playerY)) * yDeltaDist;
        }
        else
        {
            stepY = -1;
            yRayDist = Math.Abs((playerY - mapY)) * yDeltaDist;
        }
        if (playerLookXRad == 0)
            xRayDist = double.PositiveInfinity;

        if (playerLookYRad == 0)
            yRayDist = double.PositiveInfinity;

        double rayDistance = 0;
        bool hit = false;
        face = 0;

        while (!hit)
        {
            if (xRayDist < yRayDist)
            {
                xRayDist += xDeltaDist;
                rayDistance = xRayDist;
                mapX += stepX;

                face = 0;
            }
            else
            {
                yRayDist += yDeltaDist;
                rayDistance = yRayDist;
                mapY += stepY;

                face = 1;
            }

            try
            {
                hit = map[mapY, mapX] == 1;

            }
            catch (Exception)
            {
                Debug.WriteLine(mapY + ", " + mapX);
                throw;
            }
        }
        if (face == 0) rayDistance -= xDeltaDist;
        else rayDistance -= yDeltaDist;

        return rayDistance;
```

- AWSD-vel mozgunk és a jobb és bal nyilakkal mozgatjuk a kamerát, ezek változtathatóak a kódban.
- A téglalapok szélessége állítható, ezzel csökkentve a gépigényét a játéknak. Természetesen 1 pixel széles téglalapokkal néz ki jól.
- A timert is át lehet állítani, hogy másodpercenként kevesebb számítást kelljen elvégezni, természetesen ez az FPS-t fogja csökkenteni.

```
private void DrawWalls(DrawingContext drawingContext)
{
    double rectWidth = 1; //to set wall smoothness

    double rectX = 0;
    for (double rayDir = model.Player.LookAngle - fov / 2; rayDir <= model.Player.LookAngle + fov / 2; rayDir += rectWidth / size.Width * fov)
    {
        double rayLength = RayCalculator.CalculateRayLength(model.Player.PozX, model.Player.PozY, Math.Cos(GameLogic.ToRadians(rayDir)), -Math.Sin(GameLogic.ToRadians(rayDir)), model.MapMatrix, out int blockFace);

        double rectHeight = size.Height / rayLength;
        if(rectHeight > size.Height) rectHeight = size.Height;

        drawingContext.DrawRectangle(blockFace == 1 ? wallColor : sideWallColor, new Pen(Brushes.Black, 0), new Rect(rectX, size.Height / 2 - rectHeight / 2, rectWidth, rectHeight));

        rectX += rectWidth;
    }
}
```

```
public MainWindow()
{
    InitializeComponent();
    logic = new GameLogic();
    display.SetupModel(logic.model);

    Timer timer = new Timer();
    timer.Tick += Timer_Tick;
    timer.Interval = 16; //base = 1, 60fps = 16ms, 30 = 33ms, 24fps = 41ms
    timer.Start();


}
```

FPS mode

Map mode

Köszönöm a figyelmet!

Várlak a Backroomsban…