**Interpolating Polygons**

Any polygon that is partially in front of the player and partially behind the player will fall into one of two cases

* (i) One point behind the player
* (ii) One point in front of the player



1. (ii)

For Case (i):

The diagram above labels the points on the polygon as P1, P2 and P3, where P1 is behind the player and the remaining points are in front of the player. P1 is interpolated to two points on the polygon at , I2 and I3 on P1P2 and P1P3 respectively. See Interpolation method below.

The part of the polygon behind is not drawn, meaning the remaining shape will be the quadrilateral I2P2P3I3. This quadrilateral is instead drawn to the canvas (once perspective is taken into account).

For Case (ii):

The diagram above labels the points on the polygon as P1, P2 and P3, where P1 is in front of the player and the remaining points are behind the player. P1 is interpolated to two points on the polygon at , I2 and I3 on P1P2 and P1P3 respectively. See Interpolation method below.

The part of the polygon behind is not drawn, meaning the remaining shape will be the triangle I2P1I3. This triangle is instead drawn to the canvas (once perspective is taken into account).

Interpolation method

For a more general notation say the point to be changed is q1 (q1x, q1y, q1z) and the point to interpolate towards is q2 (q2x, q2y, q2z). The interpolated point to find is i (ix, iy, iz).

Notice the choice to use change in y for the gradient as

The method can be written in python as:

# returns the interpolated point on the line q1 q2 where y coordinate is gamma

# assumes y coordinates are different for q1 and q2

def interpolateUsingY(self, q1, q2):

dyFactor = 1 / (q2[1] - q1[1])

mXY = (q2[0] - q1[0]) \* dyFactor

ix = q1[0] + mXY \* (gamma - q1[1])

mZY = (q2[2] - q1[2]) \* dyFactor

iz = q1[2] + mZY \* (gamma - q1[1])

return [ix, gamma, iz]

In either case (i) or (ii) the interpolated points can be found by calling:

I2 = self.interpolateUsingY(P1, P2)

I3 = self.interpolateUsingY(P1, P3)