

1.

Merged branches, Player.cs FeetPos wasn't updating. ~~Merge~~, creating Max Branch V3 after Merging to Main should let me code collision now.

^  
It was only calling in setup()

Liam collision code: nested if

This is beyond my learnings

statements checking  
Player vs hole

X then Y, sub  
sequently

You have to call Vector2 Player Feet  
as the Method!

calling Player class calls a different instance  
of Player

FML

Practice rectangle collision is off ← Why?  
Apparently resembles this

2.

The Bad Rect Size. Y was off from  
the illustration for collision. Unclear why  
this is. ~~Will~~<sup>may</sup> need to subtract Y pos for  
collision in the future.

reshaping it's  
the .Y will need assistance lower  
than the illustration  
E God bless Liam



Float leftEdge 1 = bad Rect pos.x <sup>implement diff in x pos</sup>

→ repeat for obstacle coords

Float player Edge = feex Position.x  
→ repeat

bool doesOverlap = leftEdge 1 < player Edge  
→ repeat

if doesOverlapAll = all of those with &&'s

if (doesOverlap)

{

on pit = true;

}

← This will interact with Liam's code, killing the player.

This should work as the ~~objects~~ obstacle's collisions seem to be fixed. Liam determined it wasn't appearing as illustrated for REASONS UNKNOWN.

Removing

each verification of

doesOverlap after each new

obstacle is placed breaks the

code. solution: DON'T TOUCH!

★ Messing with

rect size.y

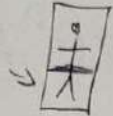
fixed it.

Position + size for right edge

COMPUTING collisions:  $\text{rectPos.X} + \text{rectSize.X}$

↑  
floats

player legs.X(ty) for primary colliding force



float PlayerEdge = joePlayer.feetPosition.X, etc.

assignment isn't ~~uploading~~ copying to bin.<sup>PP</sup> running code

to save time I'm practicing in a separate program before updating my branch

This sucks

Collision isn't working. Following the Module but Won't register

Nothing is functioning with collision.

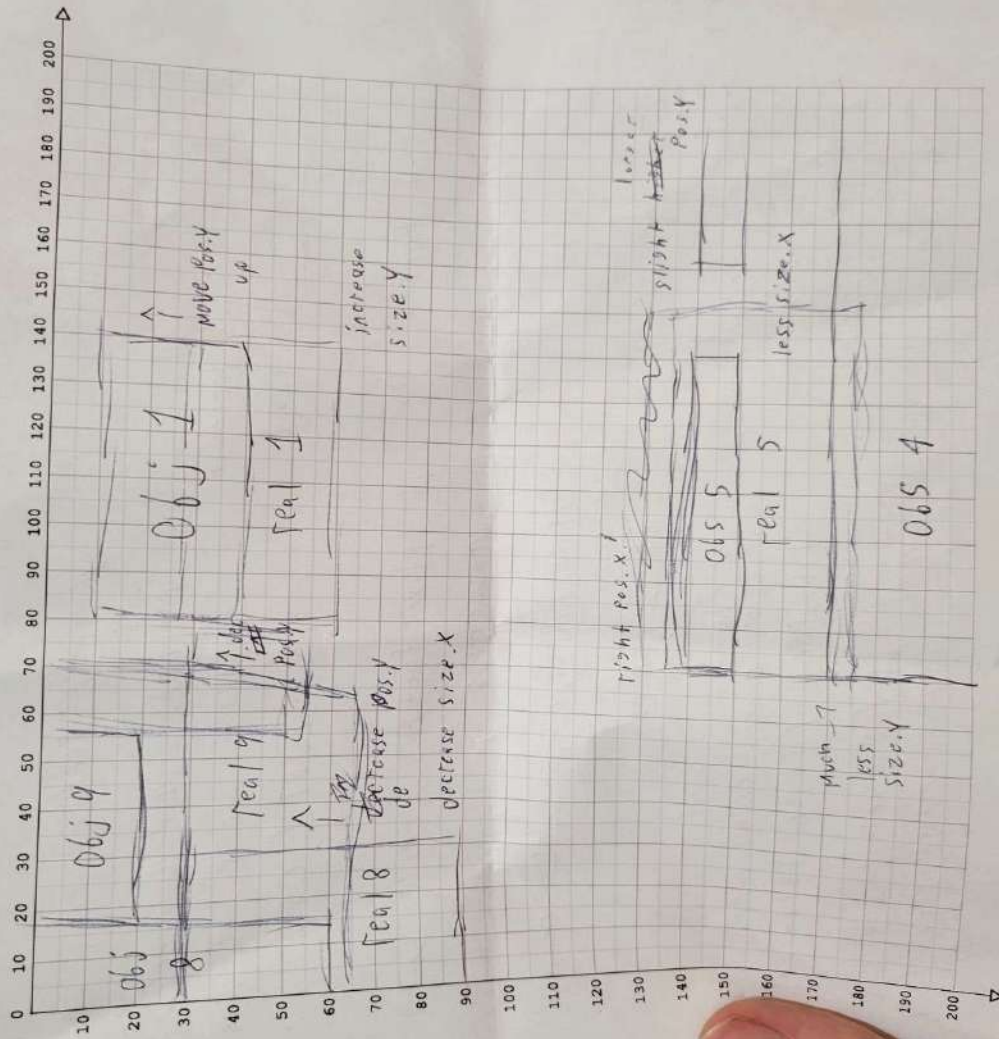
With "or" operators it'll detect incorrectly, but with "and" it just won't.

edges like this

Will check with Liam just won't register??

~~tomorrow~~ Thursday in person, implementation not possible until issue resolved

diff positions



Pixel Paper - 200 x 200  
Graph Paper for Screen Coordinates

size x = 237

size y = ~~280~~ 150

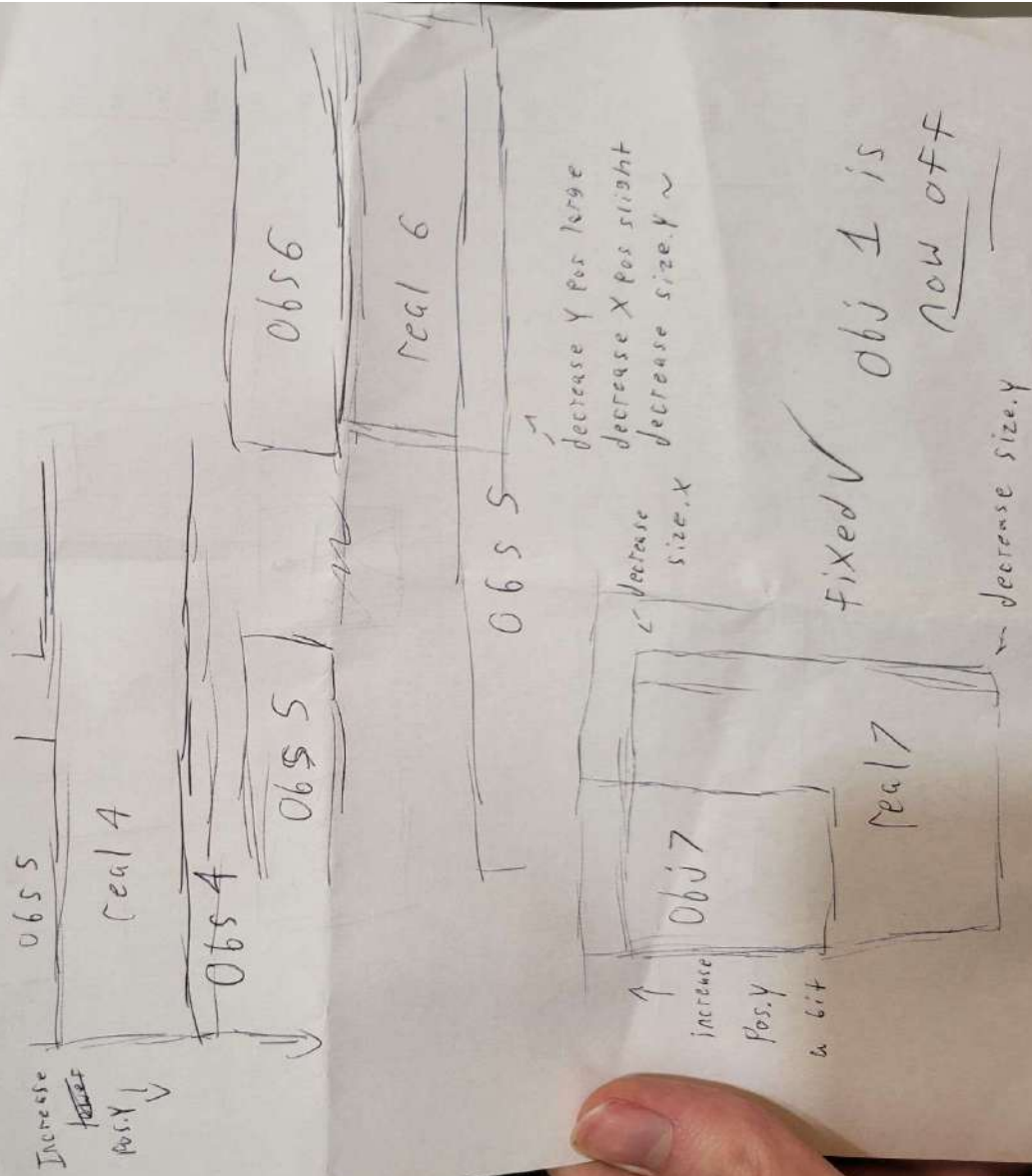
Feat pos.y = -130 instead of 30(?)

↑  
This works, it's colliding.



Obstacle 4 isn't registering for  
some reason.

It is drawing the rectangle fine,  
but it is registered in a dif position



Maybe write

Points Count --

Points Count ++

Write bool isPresent(  
collected  
= false

to make sure the process  
ends on a +1.?

Doesn't work, just increases

~~LA~~

then, in each does PresentOverlap, include isPresentCollected  
Making a && conditional that reveals the stage  
shift should work.

p3

p4

~~decrease size~~

p3

↑ decrease pos y

p4

↑ decrease  
pos. y

collision points

all seem to be too  
low in pos. y

Exactly 70 pixels too low... for  
reasons...?  
(except present #9 was 100.?)

loading textures;


we'll say "texture"

Class: Texture2D (Variable Name);

Setup: Graphics.LoadTexture ("MohawkGame2\\Images\\~~texture~~ (insert).PNG");

Update: Graphics.Draw(texture, (desired position vector));

↑ Something isn't working here either.?? ↑

Vector2 enemySpeed = new (100, 0) ○ → 

Vector2 enemy2player = player.pos - enemy.pos

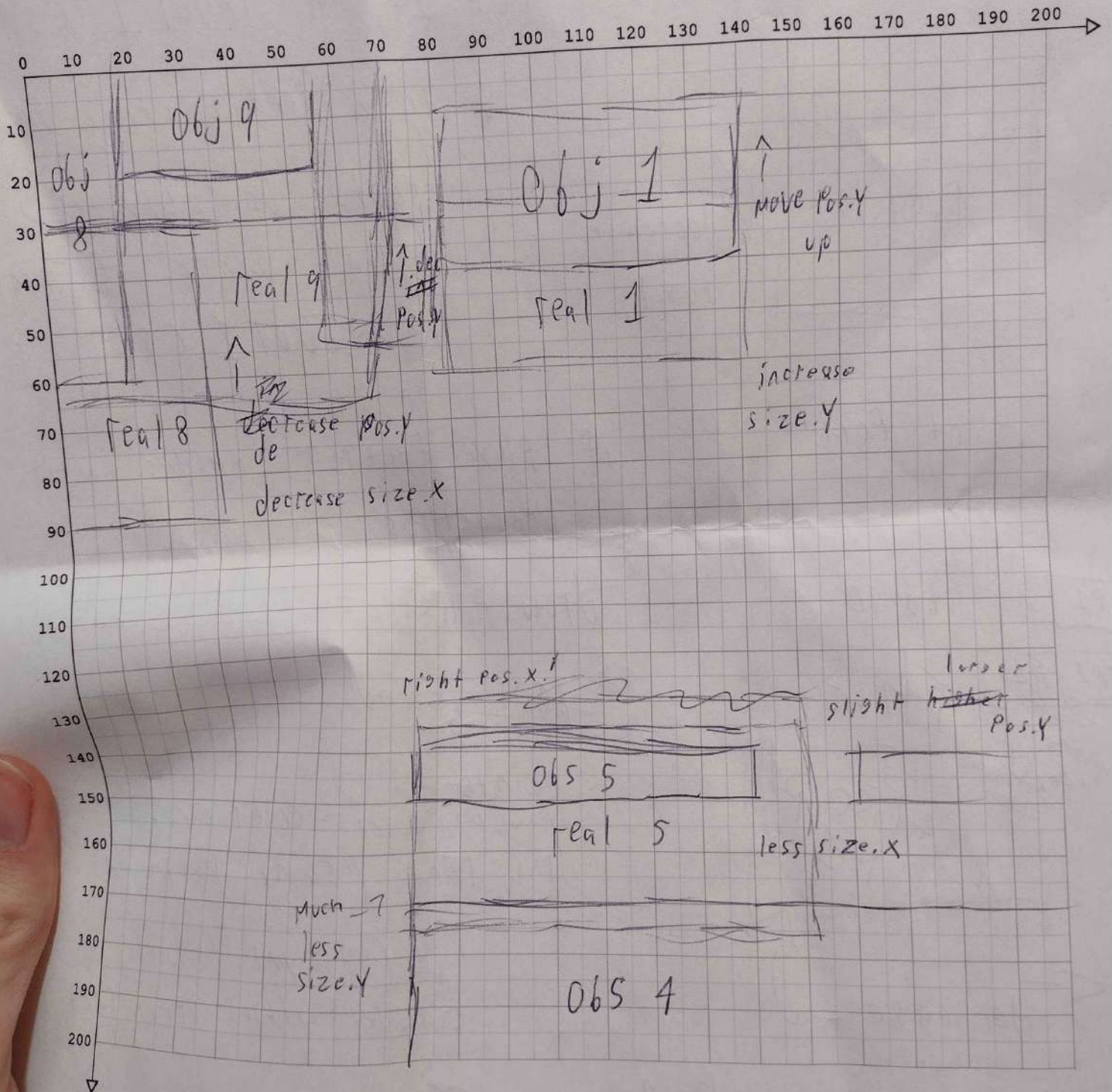
enemySpeed = Vector2.Normalize(enemy2player) \* 200

enemyPos += enemySpeed \* Time.deltaTime

↑  
stops at player starting pos.??



# diff positions



size X = 232

size Y = ~~150~~ 150

rect pos. Y = -130 instead of 30(?)

Have to remove code  
not allowing player to leave  
bounds for this space

