

1.

Merged branches, Player.cs Feet.cs Wasn't
Updating. ~~MEG~~, 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
statements checking
player vs hole

This is beyond my
learnings

X then Y sub
sequently

You have to call Vector2 playerfeet
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~~ need to subtract Y pos for collision in the future.
^{May}

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

implement diff
in x pos

float leftEdge1 = badRect pos.x ✓

repeat for obstacle coords

float playerEdge = feex position.x
repeat

bool doesOverlap = LeftEdge1 < playerEdge

repeat

if doesOverlapAll = all of those with &&'s

if (doesOverlap)

{

onpit = true;

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

}

This should work as if the ~~object~~ obstacles' collisions seem to be fixed. Liam determined it wasn't appearing as illustrated for REASONS UNKNOWN.

removing

✓ Messing with

each verification of

rect size.y

doesOverlap after each new

fixed it.

obstacle is placed breaks the

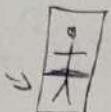
code. Solution: Don't Touch!

position + size for right edge

computing collisions: RectPos.X + RectSize.X

↑
floats

playerLegs.X(Y) for primary colliding force



float playerEdge = joePlayer.feetPosition.X, etc.

assignment isn't ~~uploading~~ copying to bin??
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.

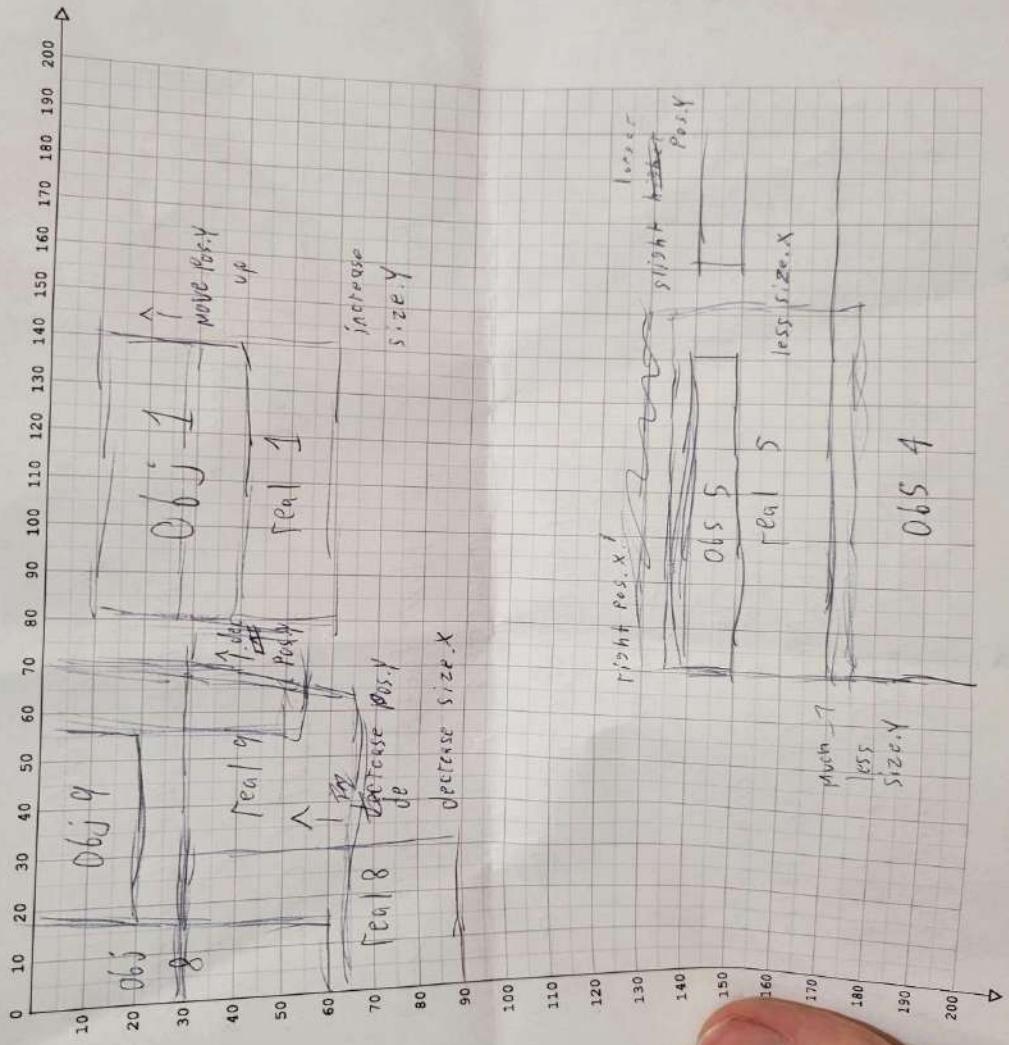
With "or" operators it'll detect incorrectly,
but with "and" it just won't.

edges like this

will check with Liam just won't register??

implementation not possible until issue resolved

diff positions



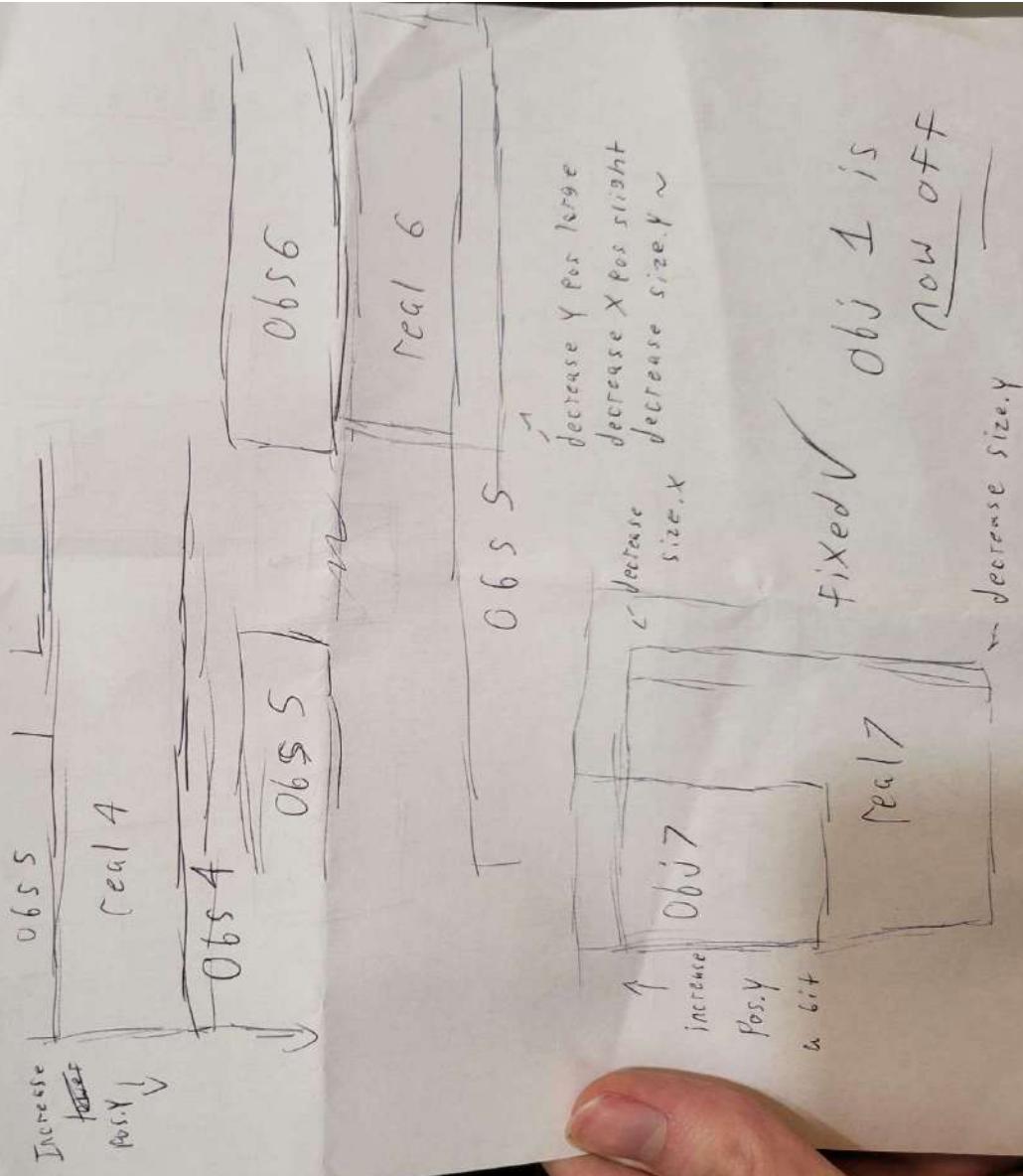
Pixel Paper - 200 x 200
Graph Paper for Screen Coordinates

size x = 25?
size y = 250
rect pos.y = 130 instead of 30(?)

1
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 diff position



Maybe write

pointsCount --

pointsCount ++

write bool isPresent(#),
collected
= false

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

Doesn't work, just increases

~~try this~~

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

[P3]

~~decrease size~~

[P3]

↑ decrease pos.Y

[P4]

[P4]

↑ decrease
pos.Y

collision points

all seem to be too
low in pos.Y

exactly 70 pixels too low... for

(except present #9 reasons...?
was 180?)

Loading textures:

We'll say "texture"

Class: Texture2D (Variable Name);

Setup: Graphics.LoadTexture ("MohawkGameZ\\IMages\\~~insert~~ (insert).PNG")

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

↑ Something isn't working here either? ↑

Vector2 enemySpeed = new (100, 0) O -> 

Vector2 enemy2Player = player.Pos - enemy.Pos

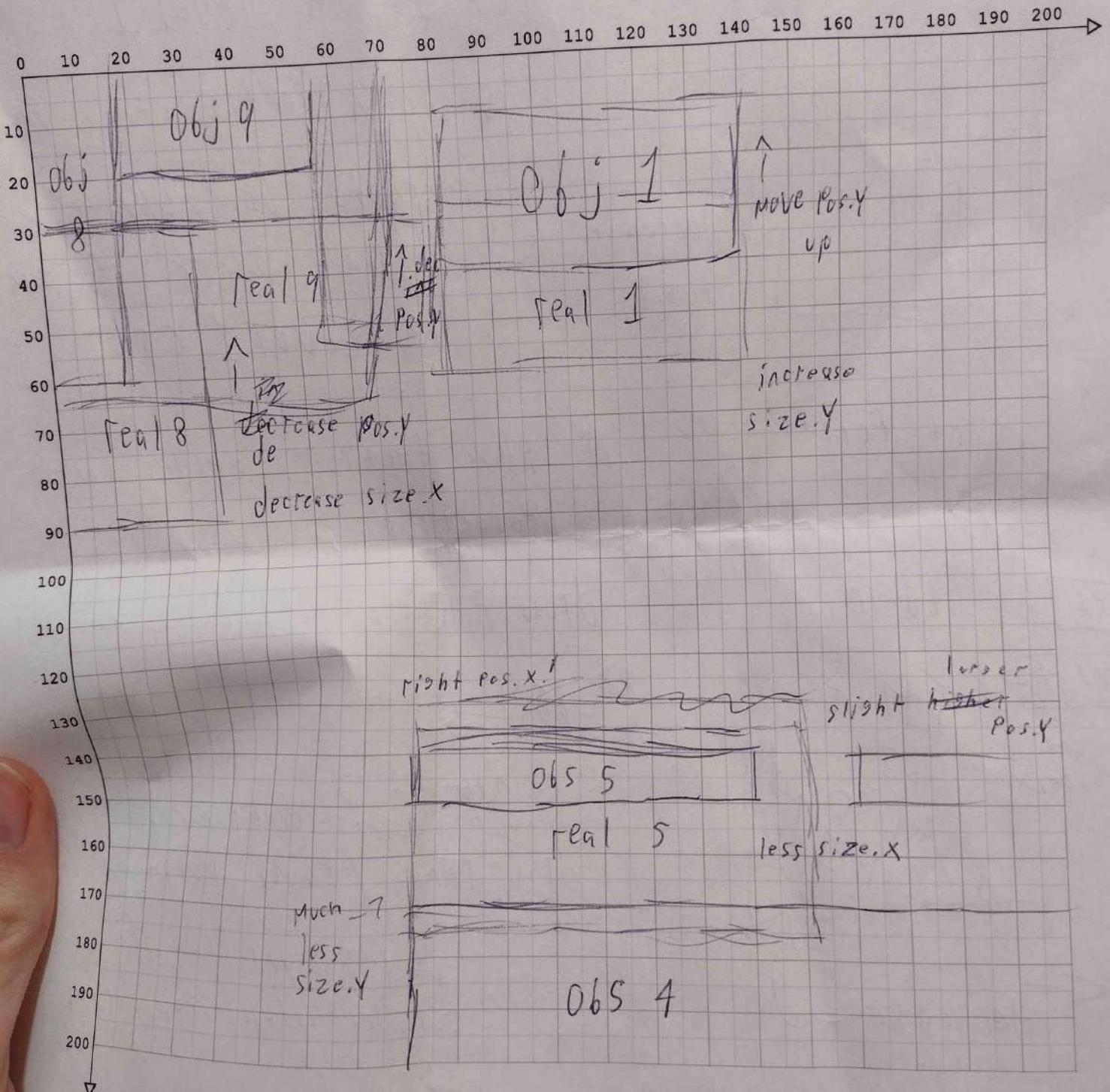
enemySpeed = Vector2.Normalize(enemy2Player) * 200

enemyPos += enemySpeed * Time.deltaTime



stops at player starting pos? ↗

diff positions



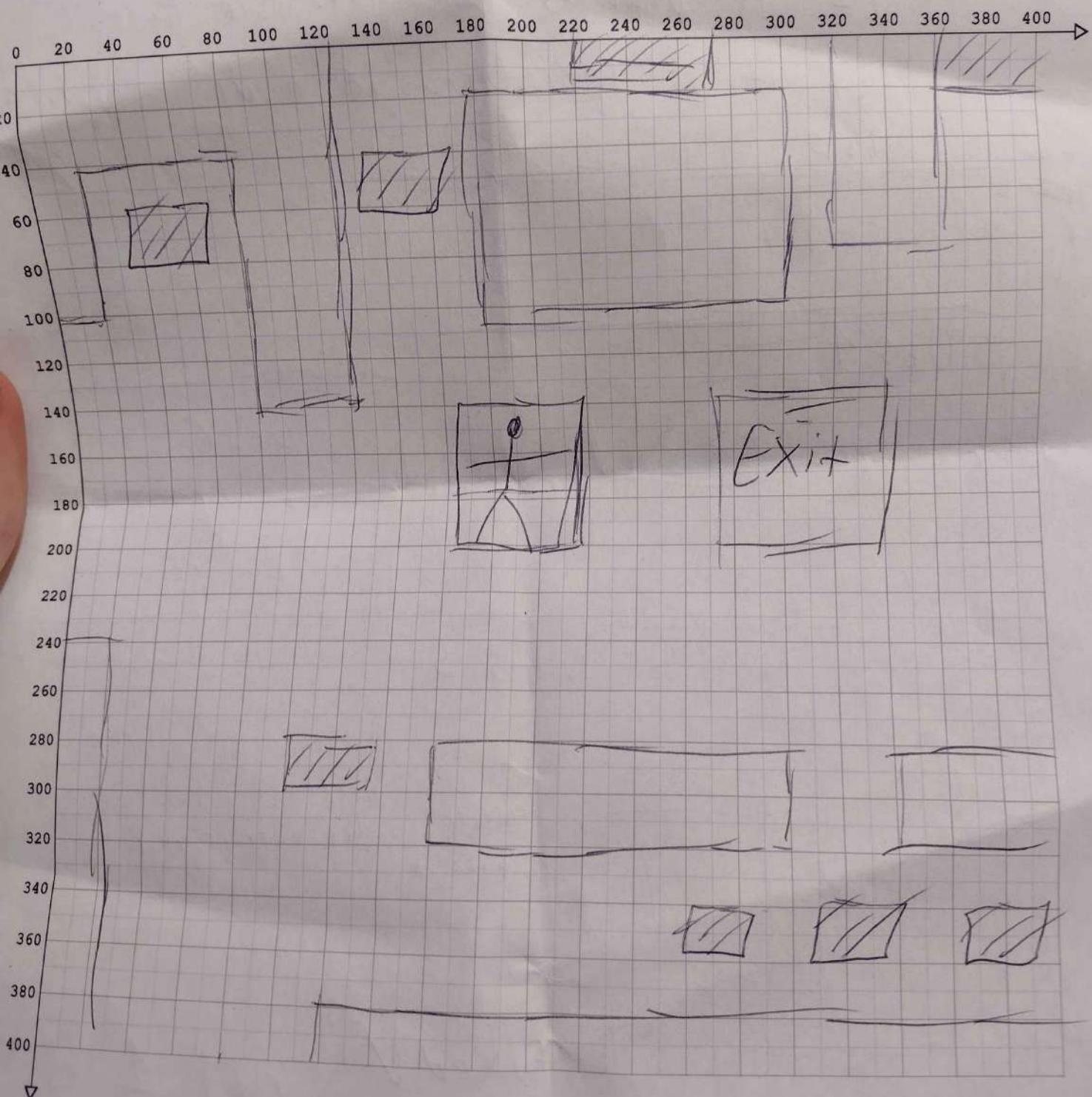
size X = 237

size Y = -150

Rect pos.y = -130 instead of 30(.)

Pixel Paper - 200 x 200
Graph Paper for Screen Coordinates

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



Pixel Paper - 400 x 400
Graph Paper for Screen Coordinates

