Pseudocode for version 3.0.0

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// feature updates

// bad sqrts are all part of version 3.1.\*

// fps is all part of version 3.2.\*

// version display is part of version 3.3.0

//Globals

//max bad squares

Bad\_sqrt\_max = number between 100 and 200

//bad square color and color being neon green for a bad bio-chemical color

Bad\_sqrt\_color = “neon green

//list for bad\_sqrts

Bad\_sqrts = []

// FPS values that are needed

start\_time = 0

frame\_rate = 30

ticks = 0

//in the create class collision functure

//for the players

for bad\_sqrt in bad\_sqrts:

if math.sqrt(((player.xPos - (WIDTH / 2) + bad\_sqrt.xPos) \*\* 2 + (player.yPos - (HEIGHT / 2) + bad\_sqrt.yPos) \*\* 2)) <= bad\_sqrt.size + player.size and bad\_sqrt.size <= player.size:

bad\_sqrts.remove(bad\_sqrt)

// This sets the takes size after bot eats a ball

player.size -= 1

// This will respawn bad sqrts

new\_sqrt = Create(random.randint(-map\_size, map\_size), random.randint(-map\_size, map\_size),bad\_sqrt\_color, 20, "sqrt")

bad\_sqrts.append(new\_sqrt)

for bot in bots:

…

Else:

…

for bad\_sqrt in bad\_sqrts:

if math.sqrt(((bot.xPos - bad\_sqrt.xPos) \*\* 2 + (bot.yPos - bad\_sqrt.yPos) \*\* 2)) <= bad\_sqrt.size + bot.size and bad\_sqrt.size <= bot.size:

bad\_sqrts.remove(bad\_sqrt)

// This sets the takes size after bot eats a ball

bot.size -= 1

// This will respawn bad sqrts

new\_sqrt = Create(random.randint(-map\_size, map\_size), random.randint(-map\_size, map\_size), bad\_sqrt\_color, 20, "sqrt")

bad\_sqrts.append(new\_sqrt)

// in create class draw had changes in 3.1.\*

If self.name != “sprt:

… stuff from version 1.0.0

Else:

// This draws the squares 3.1.0v

pygame.draw.rect(surface, self.color, (x, y, int(self.size), int(self.size)))

// in the menu loop

// to let the player know what version they are on.

Message = SMALLFONT.rendr( \*version\*, color = white)

SCREEN.blit(message, (0, HEIGHT – 30))

In the game loop

If start\_game:

…

// creates the bad\_sqrt on screen

for i in range(bad\_sqrt\_max): // and size change in 3.1.1v

new\_sqrt = Create(random.randint(-map\_size, map\_size), random.randint(-map\_size, map\_size), bad\_sqrt\_color, 20, "sqrt")

bad\_sqrts.append(new\_sqrt)

…

// for drawing the bad\_sqrts

for bad\_sqrt in bad\_sqrts:

bad\_sqrt.draw(SCREEN, bad\_sqrt.xPos + player.xPos, bad\_sqrt.yPos + player.yPos)

// for game\_over apart of the game loop

if game\_over:

…

if start\_game:

// for resetting list

bad\_sqrts = []

// for creating bad sqrts again.

for i in range(bad\_sqrt\_max): # and size change in 3.1.1v

new\_sqrt = Create(random.randint(-map\_size, map\_size), random.randint(-map\_size, map\_size), bad\_sqrt\_color, 20, "sqrt")

bad\_sqrts.append(new\_sqrt)

// game loop continues over again with FPS being right

start\_time = time.time()

//near the end of the game loop is the FPS tracking code. It keeps track of the the latency, which is maxed at 30FPS

//fps

ticks += 1

if (time.time() – startTime) > \*try 0.5 as a frame rate delay, but it may be different with a 60fps\* :

frameRate = round(counter/(time.time() -startTime))

ticks = 0

startTime = time.time()

if frameRate > 20:

message = FONT.render(“Frame Speed: “ + str(frameRate), color = green)

SCREEN.blit(message, (10, 10) )

elif framerate <= 20 and frameRate > 10:

message = FONT.render(“Frame Speed: “ + str(frameRate), color = yellow)

SCREEN.blit(message, (10, 10) )

elif framerate <= 10 and frameRate > 0:

message = FONT.render(“Frame Speed: “ + str(frameRate), color = red)

SCREEN.blit(message, (10, 10) )