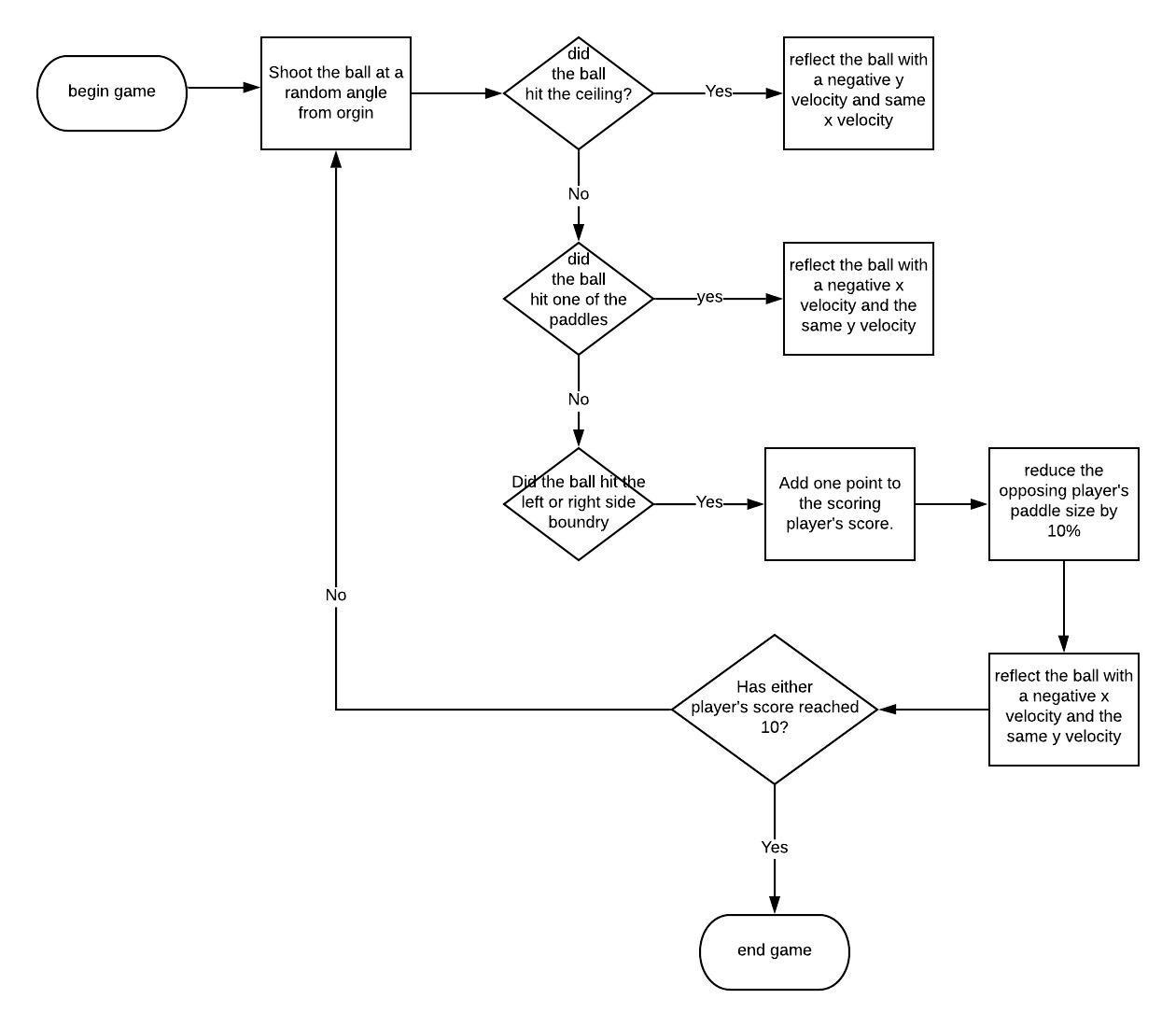
****

**import pygame**

**from paddle import Paddle**

**from ball import Ball**

**pygame.init() #intiate pygame**

**BLACK = (0,0,0)**

**WHITE = (255,255,255) #define colors**

**size = (700, 500)**

**screen = pygame.display.set\_mode(size)#intialize the bounds of the screen**

**pygame.display.set\_caption("Pong") #Create a caption**

**paddleA = Paddle(WHITE, 10, 100)**

**paddleA.rect.x = 20**

**paddleA.rect.y = 200 #create the first paddle and set it at an intial position**

**paddleB = Paddle(WHITE, 10, 100)**

**paddleB.rect.x = 670**

**paddleB.rect.y = 200 #create the second paddle and set it at an intial position**

**ball = Ball(WHITE,10,10)**

**ball.rect.x = 345**

**ball.rect.y = 195 #create the ball and set it at an intial ostion**

**all\_sprites\_list = pygame.sprite.Group() # Create a list with all of the objects**

**all\_sprites\_list.add(paddleA)**

**all\_sprites\_list.add(paddleB)**

**all\_sprites\_list.add(ball)**

**play\_verif = True #set a varible equal to true**

**clock = pygame.time.Clock()**

**scoreA = 0**

**scoreB = 0 #intialize scores**

**while play\_verif:**

**for event in pygame.event.get():**

**if event.type == pygame.QUIT:**

**play\_verif = False**

**elif event.type==pygame.KEYDOWN:**

**if event.key==pygame.K\_x:**

**play\_verif = False # quit the game when user presses x**

**keys = pygame.key.get\_pressed()**

**if keys[pygame.K\_w]:**

**paddleA.moveUp(5)**

**if keys[pygame.K\_s]:**

**paddleA.moveDown(5)# when w is pressed move paddle a up when s is pressed move paddle a down**

**if keys[pygame.K\_UP]:**

**paddleB.moveUp(5)**

**if keys[pygame.K\_DOWN]:**

**paddleB.moveDown(5)# when up key is pressed move paddle b up when down key is pressed move paddle b down**

**all\_sprites\_list.update()**

**if ball.rect.x>=690:**

**ball.velocity[0] = -ball.velocity[0]**

**scoreA+=1**

**if ball.rect.x<=0:**

**ball.velocity[0] = -ball.velocity[0]**

**if ball.rect.y>490:**

**scoreB+=1**

**ball.velocity[1] = -ball.velocity[1]**

**if ball.rect.y<0:**

**ball.velocity[1] = -ball.velocity[1] #Have the ball bounce of all surfaces and add 1 to opposing teams score if it hits side wall**

**if pygame.sprite.collide\_mask(ball, paddleA) or pygame.sprite.collide\_mask(ball, paddleB):**

**ball.bounce() #If the ball hits either paddle bounce it off**

**screen.fill(BLACK)**

**screen.fill(BLACK)**

**pygame.draw.line(screen, WHITE, [349, 0], [349, 500], 5)**

**all\_sprites\_list.draw(screen)**

**font = pygame.font.Font(None, 74)**

**text = font.render(str(scoreA), 1, WHITE)**

**screen.blit(text, (250,10))**

**text = font.render(str(scoreB), 1, WHITE)**

**screen.blit(text, (420,10)) #The setting for the text that displays score**

**pygame.display.flip()**

**clock.tick(60)**

**pygame.quit()**

**pygame.display.flip()**

**clock.tick(60)**

**pygame.quit()**

**import pygame**

**from random import randint**

**BLACK = (0,0,0)**

**class Ball(pygame.sprite.Sprite):**

**def \_\_init\_\_(self, color, width, height):**

**super().\_\_init\_\_()**

**self.image = pygame.Surface([width, height])**

**self.image.fill(BLACK)**

**self.image.set\_colorkey(BLACK)**

**pygame.draw.rect(self.image, color, [0, 0, width, height])**

**self.velocity = [randint(4,8),randint(-8,8)]**

**self.rect = self.image.get\_rect()**

**def update(self):**

**self.rect.x += self.velocity[0]**

**self.rect.y += self.velocity[1]**

**def bounce(self):**

**self.velocity[0] = -self.velocity[0]**

**self.velocity[1] = randint(-8,8) #The function that defines how to bounce the ball**

**import pygame**

**BLACK = (0,0,0)**

**class Paddle(pygame.sprite.Sprite):**

**def \_\_init\_\_(self, color, width, height):**

**super().\_\_init\_\_()**

**self.image = pygame.Surface([width, height])**

**self.image.fill(BLACK)**

**self.image.set\_colorkey(BLACK)**

**pygame.draw.rect(self.image, color, [0, 0, width, height])**

**self.rect = self.image.get\_rect()**

**def moveUp(self, pixels):**

**self.rect.y -= pixels**

**if self.rect.y < 0:**

**self.rect.y = 0 #function to move the paddle up**

**def moveDown(self, pixels):**

**self.rect.y += pixels**

**if self.rect.y > 400:**

**self.rect.y = 400 #function to move the paddle down**

Citations

“Python Game Tutuorial.” *Youtube.com*, FreecodeCamp.com, www.youtube.com/watch?v=C6jJg9Zan7w.