1) main.py

```
# Import the pygame library and initialise the game engine
import pygame
pygame.init()

# Define some colors
BLACK = (0,0,0)
WHITE = (255,255,255)

# Open a new window
size = (700, 500)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Pong")

# The loop will carry on until the user exits the game (e.g. clicks the close button).
carryOn = True

# The clock will be used to control how fast the screen updates
clock = pygame.time.Clock()

# -------- Main Program Loop -----------
while carryOn:
    # --- Main event loop
    for event in pygame.event.get(): # User did something
        if event.type == pygame.QUIT: # If user clicked close
              carryOn = False # Flag that we are done so we exit this loop

    # --- Game logic should go here



    # --- Drawing code should go here
    # First, clear the screen to black.
    screen.fill(BLACK)
    #Draw the net
    pygame.draw.line(screen, WHITE, [349, 0], [349, 500], 5)

    # --- Go ahead and update the screen with what we've drawn.
    pygame.display.flip()

    # --- Limit to 60 frames per second
    clock.tick(60)

#Once we have exited the main program loop we can stop the game engine:
pygame.quit()
```

2) paddle.py

```
import pygame
BLACK = (0,0,0)

class Paddle(pygame.sprite.Sprite):
    #This class represents a paddle. It derives from the "Sprite" class in Pygame.

    def __init__(self, color, width, height):
        # Call the parent class (Sprite) constructor
        super().__init__()

        # Pass in the color of the paddle, its width and height.
        # Set the background color and set it to be transparent
        self.image = pygame.Surface([width, height])
        self.image.fill(BLACK)
        self.image.set_colorkey(BLACK)

        # Draw the paddle (a rectangle!)
        pygame.draw.rect(self.image, color, [0, 0, width, height])

        # Fetch the rectangle object that has the dimensions of the image.
        self.rect = self.image.get_rect()
```

3) main.py

```
# Import the pygame library and initialise the game engine
import pygame
from paddle import Paddle

pygame.init()

# Define some colors
BLACK = (0,0,0)
WHITE = (255,255,255)

# Open a new window
size = (700, 500)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Pong")

paddleA = Paddle(WHITE, 10, 100)
paddleA.rect.x = 20
paddleA.rect.y = 200

paddleB = Paddle(WHITE, 10, 100)
paddleB.rect.x = 670
paddleB.rect.y = 200

#This will be a list that will contain all the sprites we intend to use in our game.
all_sprites_list = pygame.sprite.Group()

# Add the paddles to the list of sprites
all_sprites_list.add(paddleA)
all_sprites_list.add(paddleB)

# The loop will carry on until the user exits the game (e.g. clicks the close button).
carryOn = True

# The clock will be used to control how fast the screen updates
clock = pygame.time.Clock()

```

```
36
37  # -------- Main Program Loop -----------
38  while carryOn:
39      # --- Main event loop
40      for event in pygame.event.get(): # User did something
41          if event.type == pygame.QUIT: # If user clicked close
42                carryOn = False # Flag that we are done so we exit this loop
43          elif event.type==pygame.KEYDOWN:
44                  if event.key==pygame.K_x: #Pressing the x Key will quit the game
45                       carryOn=False
46
47      # --- Game logic should go here
48      all_sprites_list.update()
49
50
51      # --- Drawing code should go here
52      # First, clear the screen to black.
53      screen.fill(BLACK)
54      #Draw the net
55      pygame.draw.line(screen, WHITE, [349, 0], [349, 500], 5)
56
57      #Now let's draw all the sprites in one go. (For now we only have 2 sprites!)
58      all_sprites_list.draw(screen)
59
60      # --- Go ahead and update the screen with what we've drawn.
61      pygame.display.flip()
62
63      # --- Limit to 60 frames per second
64      clock.tick(60)
65
66  #Once we have exited the main program loop we can stop the game engine:
67  pygame.quit()
```