

PONG GAME

In [10]:

```
import turtle
```

In [11]:

```
#Window
wind = turtle.Screen()
wind.title('CTXGO')
wind.bgcolor('black')
wind.setup(width=800, height=600)
wind.tracer(0)
```

In [12]:

```
#Bar A
bar_A = turtle.Turtle()
bar_A.shape('square')
bar_A.color('grey')
bar_A.shapesize(stretch_wid=5, stretch_len=1)
bar_A.penup()
bar_A.goto(-350,0)
```

In [13]:

```
# Bar B
bar_B = turtle.Turtle()
bar_B.shape('square')
bar_B.color('grey')
bar_B.shapesize(stretch_wid=5, stretch_len=1)
bar_B.penup()
bar_B.goto(350,0)
```

In [14]:

```
# Ball
ball = turtle.Turtle()
ball.shape('circle')
ball.color('grey')
ball.penup()
ball.goto(0,0)
ball_x = 0.1
ball_y = 0.1
```

In [15]:

```
#score
sboard = turtle.Turtle()
sboard.shape('square')
sboard.color('grey')
sboard.penup()
sboard.hideturtle()
sboard.goto(0, 260)
sboard.write("Player A: 0 Player B: 0", align="center", font=("Courier", 24,'normal'))

score_a = 0
score_b = 0
```

In [16]:

```
# Functions
def bar_A_up():
    y = bar_A.ycor()
    y += 30
    bar_A.sety(y)
def bar_A_down():
    y = bar_A.ycor()
    y -= 30
    bar_A.sety(y)
def bar_B_up():
    y = bar_B.ycor()
    y += 30
    bar_B.sety(y)
def bar_B_down():
    y = bar_B.ycor()
    y -= 30
    bar_B.sety(y)
```

In [17]:

```
# Keyboard Bindings
wind.listen()
wind.onkeypress(bar_A_up, 'w')
wind.onkeypress(bar_A_down, 's')
wind.onkeypress(bar_B_up, 'Up')
wind.onkeypress(bar_B_down, 'Down')
```

In [18]:

```
ball_speed_x=7
ball_speed_y=7
```

In [19]:

```

while True:
    wind.update()

    # BALL movement
    ball.setx(ball.xcor() + ball_x)
    ball.sety(ball.ycor() + ball_y)

    # Border
    if ball.ycor() > 290:
        ball.sety(290)
        ball_y *= -1
    elif ball.ycor() < -290:
        ball.sety(-290)
        ball_y *= -1

    #score
    if ball.xcor() > 350:
        score_a += 1
        sboard.clear()
        sboard.write("Player A: {} Player B {}".format(score_a, score_b), align='center', f
        ball.goto(0,0)
        ball_x *= -1
    elif ball.xcor() < -350:
        score_b += 1
        sboard.clear()
        sboard.write("Player A: {} Player B {}".format(score_a, score_b), align='center',
                      font=('Courier', 24, 'normal'))
        ball.goto(0, 0)
        ball_x *= -1
    # Collision with bars
    if ball.xcor() < -340 and ball.ycor() < bar_A.ycor() + 50 and ball.ycor() > bar_A.ycor(
        ball_x *= -1
    elif ball.xcor() > 340 and ball.ycor() < bar_B.ycor() + 50 and ball.ycor() > bar_B.ycor(
        ball_x *= -1

```

Terminator

Traceback (most recent call las

t)

<ipython-input-19-e19cec414800> in <module>

```

1 while True:
----> 2     wind.update()
3
4     # BALL movement
5     ball.setx(ball.xcor() + ball_x)

```

~\anaconda3\lib\turtle.py in update(self)

```

1301         self._tracing = True
1302         for t in self.turtles():
-> 1303             t._update_data()
1304             t._drawturtle()
1305         self._tracing = tracing

```

~\anaconda3\lib\turtle.py in _update_data(self)

```

2644
2645     def _update_data(self):
-> 2646         self.screen._incrementudc()

```

```
2647         if self.screen._updatecounter != 0:
2648             return

~\anaconda3\lib\turtle.py in _incrementudc(self)
1290         if not TurtleScreen._RUNNING:
1291             TurtleScreen._RUNNING = True
-> 1292         raise Terminator
1293     if self._tracing > 0:
1294         self._updatecounter += 1
```

Terminator:

In []:

This code is for background sound

In []:

In [1]:

```
import turtle
import winsound
```

In [2]:

```
wn = turtle.Screen()
wn.title('Pong Game - Prathyusha')
wn.bgcolor('turquoise')
wn.setup(width=800, height=600)
wn.tracer(0)
```

In [3]:

```
# Paddle A
paddle_a = turtle.Turtle()
paddle_a.speed(0)
paddle_a.shape('square')
paddle_a.color('grey')
paddle_a.penup()
paddle_a.goto(-350, 0)
paddle_a.shapesize(5, 1)
```

In [4]:

```
# Paddle B
paddle_b = turtle.Turtle()
paddle_b.speed(0)
paddle_b.shape('square')
paddle_b.color('grey')
paddle_b.penup()
paddle_b.goto(350, 0)
paddle_b.shapesize(5, 1)
```

In [5]:

```
# Ball
ball = turtle.Turtle()
ball.speed(0)
ball.shape('circle')
ball.color('red')
ball.penup()
ball.dx = 0.3
ball.dy = 0.3
```

In [6]:

```
# Pen
pen = turtle.Turtle()
pen.speed(0)
pen.color('grey')
pen.penup()
pen.goto(0, 260)
pen.write("Player A: 0   Player B: 0", align='center', font=('Courier', 24, 'bold'))
pen.hideturtle()
```

In [7]:

```
# Score
score_a = 0
score_b = 0

def paddle_a_up():
    y = paddle_a.ycor()
    y += 20
    paddle_a.sety(y)

def paddle_b_up():
    y = paddle_b.ycor()
    y += 20
    paddle_b.sety(y)

def paddle_a_down():
    y = paddle_a.ycor()
    y += -20
    paddle_a.sety(y)

def paddle_b_down():
    y = paddle_b.ycor()
    y += -20
    paddle_b.sety(y)
```

In [8]:

```
# Keyboard binding
wn.listen()
wn.onkeypress(paddle_a_up, 'w')
wn.onkeypress(paddle_a_down, 's')
wn.onkeypress(paddle_b_up, 'Up')
wn.onkeypress(paddle_b_down, 'Down')
```

In [9]:

```

# Main game loop
while True:
    wn.update()

    # Moving Ball
    ball.setx(ball.xcor() + ball.dx)
    ball.sety(ball.ycor() + ball.dy)

    # Border checking
    if ball.ycor() > 290 or ball.ycor() < -290:
        winsound.PlaySound('bounce.wav', winsound.SND_ASYNC)
        ball.dy *= -1

    if ball.xcor() > 390:
        winsound.PlaySound('bounce.wav', winsound.SND_ASYNC)
        ball.goto(0, 0)
        ball.dx *= -1
        score_a += 1
        pen.clear()
        pen.write("Player A: {} Player B: {}".format(score_a, score_b), align='center', fo

    if ball.xcor() < -390:
        winsound.PlaySound('bounce.wav', winsound.SND_ASYNC)
        ball.goto(0, 0)

        ball.dx *= -1
        score_b += 1
        pen.clear()
        pen.write("Player A: {} Player B: {}".format(score_a, score_b), align='center', fo

    # Paddle and ball collisions
    if (ball.xcor() > 340 and ball.xcor() < 350) and (ball.ycor() < paddle_b.ycor() + 60 an
        winsound.PlaySound('bounce.wav', winsound.SND_ASYNC)
        ball.setx(340)
        ball.dx *= -1

    if (ball.xcor() < -340 and ball.xcor() > -350) and (ball.ycor() < paddle_a.ycor() + 60
        winsound.PlaySound('bounce.wav', winsound.SND_ASYNC)
        ball.setx(-340)
        ball.dx *= -1

```

-
TclError

Traceback (most recent call las

t)

<ipython-input-9-7548d3c4085b> in <module>

4

5 # Moving Ball

----> 6 ball.setx(ball.xcor() + ball.dx)

7 ball.sety(ball.ycor() + ball.dy)

8

~\anaconda3\lib\turtle.py in setx(self, x)

1806 (10.00, 240.00)

1807 """

-> 1808 self._goto(Vec2D(x, self._position[1]))

1809

1810 def sety(self, y):

```
~\anaconda3\lib\turtle.py in _goto(self, end)
    3156         (self.currentLineItem,
    3157         self.currentLine[:],
-> 3158         screen._pointlist(self.currentLineItem),
    3159         self.items[:])
    3160     )

~\anaconda3\lib\turtle.py in _pointlist(self, item)
    753         (9.999999999999982, 0.0)]
    754     >>> """
-> 755     cl = self.cv.coords(item)
    756     pl = [(cl[i], -cl[i+1]) for i in range(0, len(cl), 2)]
    757     return pl

<string> in coords(self, *args, **kw)

~\anaconda3\lib\tkinter\__init__.py in coords(self, *args)
    2759         return [self.tk.getdouble(x) for x in
    2760                 self.tk.splitlist(
-> 2761                     self.tk.call((self._w, 'coords') + args))]
    2762
    2763     def _create(self, itemType, args, kw): # Args: (val, val, ...,
cnf={})

TclError: invalid command name "!.canvas"
```

In []:

In []: