Day - 22

**Pong: The Famous Arcade Game**

Project building, Break down the problem.

**22.1 Break the problem on your own way**

|  |  |
| --- | --- |
| Practice | Instructor |
| 1. Create Screen, 2. Create paddle 3. Move paddle 4. Create ball 5. Move Ball 6. Collision and End game 7. Divider 8. Score |  |

**22.2 Set up the Main Screen**

**import** turtle

scr\_een = turtle**.Screen**()

#*Backgrund color : "#a8c64e" and Turtle color: "#3c412c"*

scr\_een**.title**("PONG")

scr\_een**.bgcolor**("#a8c64e")

scr\_een**.setup**(width= 800, height= 600)

scr\_een**.exitonclick**()

#*python pong\_screen.py*

**22.3 Create a Paddle that responds to Key Presses**

|  |  |
| --- | --- |
| SINGLE turtle: Instructor | ARRAY turtle: Practice |
| **import** turtle  scr\_een = turtle**.Screen**()  #*Backgrund color : "#a8c64e" and Turtle color: "#3c412c"*  scr\_een**.title**("PONG")  scr\_een**.bgcolor**("#a8c64e")  scr\_een**.setup**(width= 800, height= 600)  #*animation modification*  scr\_een**.tracer**(0)  #*paddle setting*  paddle = turtle**.Turtle**(shape="square")  paddle**.shapesize**(stretch\_len=1, stretch\_wid=5)  paddle**.speed**("fastest")  paddle**.penup**()  paddle**.color**("#3c412c")  paddle**.goto**(350, 0)    **def** **change\_pos**(step):      x\_crd = paddle**.xcor**()      y\_crd = paddle**.ycor**()      paddle**.goto**(x\_crd, y\_crd + step)  **def** **up**():  **change\_pos**(-20)  **def** **down**():  **change\_pos**(20)  scr\_een**.listen**()  scr\_een**.onkeypress**(up, "Up")  scr\_een**.onkeypress**(down, "Down")  gam\_is\_on = **True**  **while** gam\_is\_on:      scr\_een**.update**()  scr\_een**.exitonclick**()  #*python pong\_screen.py*  #*ball move*  #*ball.goto(x\_crd + step, y\_crd + step)* | **import** turtle  scr\_een = turtle**.Screen**()  #*Backgrund color : "#a8c64e" and Turtle color: "#3c412c"*  scr\_een**.title**("PONG")  scr\_een**.bgcolor**("#a8c64e")  scr\_een**.setup**(width= 800, height= 600)  #*animation modification*  scr\_een**.tracer**(0)  #*paddle setting*  paddle = []  **for** i **in** **range**(0, 5):      segment = turtle**.Turtle**(shape="square")      segment**.speed**("fastest")      segment**.penup**()      segment**.color**("#3c412c")      segment**.goto**(350, -40 + (20\*i))      paddle**.append**(segment)  scr\_een**.update**()  **def** **change\_pos**(step):  **for** i **in** **range**(0, 5):          x\_crd = paddle[i]**.xcor**()          y\_crd = paddle[i]**.ycor**()          paddle[i]**.goto**(x\_crd, y\_crd + step)  **def** **up**():  **change\_pos**(-5)      scr\_een**.update**()  **def** **down**():  **change\_pos**(5)      scr\_een**.update**()  scr\_een**.listen**()  scr\_een**.onkeypress**(up, "Up")  scr\_een**.onkeypress**(down, "Down")    scr\_een**.exitonclick**()  #*python pong\_screen.py*  #*ball move*  #*ball.goto(x\_crd + step, y\_crd + step)* |

**22.4 Write the Paddle Class and Create the Second Paddle**

|  |  |
| --- | --- |
| Practice | Instructor |
| **import** turtle  STEP = 20  **class** Paddle:  **def** **\_\_init\_\_**(self, x\_coord, y\_coord):          #*paddle setting*  **self.**paddle = turtle**.Turtle**(shape="square")  **self.**paddle**.shapesize**(stretch\_len=1, stretch\_wid=5)  **self.**paddle**.speed**("fastest")  **self.**paddle**.penup**()  **self.**paddle**.color**("#3c412c")  **self.**paddle**.goto**(x\_coord, y\_coord)    **def** **change\_pos**(self, step):          x\_crd = **self.**paddle**.xcor**()          y\_crd = **self.**paddle**.ycor**()  **self.**paddle**.goto**(x\_crd, y\_crd + step)  **def** **up**(self):  **self.change\_pos**(-STEP)  **def** **down**(self):  **self.change\_pos**(STEP)  #*python pong\_paddle.py*  **import** turtle  **import** pong\_paddle  scr\_een = turtle**.Screen**()  #*Backgrund color : "#a8c64e" and Turtle color: "#3c412c"*  scr\_een**.title**("PONG")  scr\_een**.bgcolor**("#a8c64e")  scr\_een**.setup**(width= 800, height= 600)  #*animation modification*  scr\_een**.tracer**(0)  #*Paddle setup*  r\_paddle = pong\_paddle**.Paddle**(x\_coord= 350, y\_coord= 0)  l\_paddle = pong\_paddle**.Paddle**(x\_coord= -350, y\_coord= 0)  scr\_een**.listen**()  scr\_een**.onkeypress**(r\_paddle**.**up, "Up")  scr\_een**.onkeypress**(r\_paddle**.**down, "Down")  scr\_een**.onkeypress**(l\_paddle**.**up, "Left")  scr\_een**.onkeypress**(l\_paddle**.**down, "Right")  gam\_is\_on = **True**  **while** gam\_is\_on:      scr\_een**.update**()  scr\_een**.exitonclick**()  #*python pong\_screen.py*  #*ball move*  #*ball.goto(x\_crd + step, y\_crd + step)* | **import** turtle  STEP = 20  **class** Paddle(turtle**.**Turtle):  **def** **\_\_init\_\_**(self, posITon):  **super**()**.\_\_init\_\_**() #*Always use ()*          #*paddle setting*  **self.shape**("square")  **self.shapesize**(stretch\_len=1, stretch\_wid=5)  **self.speed**("fastest")  **self.penup**()  **self.color**("#3c412c")  **self.goto**(posITon)    **def** **change\_pos**(self, step):          x\_crd = **self.xcor**()          y\_crd = **self.ycor**()  **self.goto**(x\_crd, y\_crd + step)  **def** **up**(self):  **self.change\_pos**(-STEP)  **def** **down**(self):  **self.change\_pos**(STEP)  #*python pong\_paddle.py*  **import** turtle  **import** pong\_paddle  scr\_een = turtle**.Screen**()  #*Backgrund color : "#a8c64e" and Turtle color: "#3c412c"*  scr\_een**.title**("PONG")  scr\_een**.bgcolor**("#a8c64e")  scr\_een**.setup**(width= 800, height= 600)  #*animation modification*  scr\_een**.tracer**(0)  #*Paddle setup*  #*Notice Tuple is used*  r\_paddle = pong\_paddle**.Paddle**((350, 0))  l\_paddle = pong\_paddle**.Paddle**((-350, 0))  scr\_een**.listen**()  scr\_een**.onkeypress**(r\_paddle**.**up, "Up")  scr\_een**.onkeypress**(r\_paddle**.**down, "Down")  scr\_een**.onkeypress**(l\_paddle**.**up, "Left")  scr\_een**.onkeypress**(l\_paddle**.**down, "Right")  gam\_is\_on = **True**  **while** gam\_is\_on:      scr\_een**.update**()  scr\_een**.exitonclick**()  #*python pong\_screen.py*  #*ball move*  #*ball.goto(x\_crd + step, y\_crd + step)* |

**22.5 Write the Ball Class and Make the Ball Move**

|  |  |
| --- | --- |
| Practice | |
| **import** turtle  **class** Ball(turtle**.**Turtle):  **def** **\_\_init\_\_**(self, posITon):  **super**()**.\_\_init\_\_**() #*Always use ()*          #*paddle setting*  **self.shape**("circle")  **self.speed**("fastest")  **self.penup**()  **self.color**("#3c412c")  **self.goto**(posITon)  #*python pong\_ball.py.py* | **import** pong\_ball  **import** time  . . . . .  #*Ball setup. Use tuple for position*  ball = pong\_ball**.Ball**((0, 0))  gam\_is\_on = **True**  **while** gam\_is\_on:      scr\_een**.update**()      time**.sleep**(0.1)      ball**.goto**(ball**.xcor**() + 10, ball**.ycor**() + 10)  #*python pong\_main.py* |

|  |  |
| --- | --- |
| Instructor | |
| **import** turtle  **class** Ball(turtle**.**Turtle):  **def** **\_\_init\_\_**(self):  **super**()**.\_\_init\_\_**() #*Always use ()*          #*paddle setting*  **self.shape**("circle")  **self.speed**("fastest")  **self.penup**()  **self.color**("#3c412c")  **def** **move**(self):          new\_x = **self.xcor**() + 10          new\_y = **self.ycor**() + 10  **self.goto**(new\_x, new\_y)  #*python pong\_ball.py.py* | **import** pong\_ball  **import** time  . . . . .  #*Ball setup. Use tuple for position*  ball = pong\_ball**.Ball**()  gam\_is\_on = **True**  **while** gam\_is\_on:      scr\_een**.update**()      time**.sleep**(0.1)      ball**.move**()    #*python pong\_main.py* |

**22.6 Add the Ball Bouncing Logic**

|  |  |
| --- | --- |
| Practice | Instructor |
| **import** turtle  **class** Ball(turtle**.**Turtle):  **def** **\_\_init\_\_**(self):  **self.**step\_x = 10  **self.**step\_y = 10  **super**()**.\_\_init\_\_**() #*Always use ()*          #*paddle setting*  **self.shape**("circle")  **self.speed**("fastest")  **self.penup**()  **self.color**("#3c412c")  **def** **wall\_coolision**(self):  **if** **self.ycor**() **>=** 280:  **self.**step\_y = -10  **else**:  **pass**  **def** **move**(self):          new\_x = **self.xcor**() + **self.**step\_x          new\_y = **self.ycor**() + **self.**step\_y  **self.wall\_coolision**()  **self.goto**(new\_x, new\_y)  #*python pong\_ball.py* | **import** turtle  **class** Ball(turtle**.**Turtle):  **def** **\_\_init\_\_**(self):  **self.**step\_x = 10  **self.**step\_y = 10  **super**()**.\_\_init\_\_**() #*Always use ()*          #*paddle setting*  **self.shape**("circle")  **self.speed**("fastest")  **self.penup**()  **self.color**("#3c412c")  **def** **bounce**(self):  **if** (**self.ycor**() **>=** 280) **or** (**self.ycor**() **<=** -280):  **self.**step\_y \*= -1  **def** **move**(self):          new\_x = **self.xcor**() + **self.**step\_x          new\_y = **self.ycor**() + **self.**step\_y  **self.goto**(new\_x, new\_y)  #*python pong\_ball.py.py*  gam\_is\_on = **True**  **while** gam\_is\_on:      scr\_een**.update**()      time**.sleep**(0.1)  **if** (ball**.ycor**() **>=** 280) **or** (ball**.ycor**() **<=** -280):          ball**.bounce**()      ball**.move**()  scr\_een**.exitonclick**()  #*python pong\_main.py* |

**22.7 How to Detect Collisions with the Paddle**

|  |  |
| --- | --- |
| gam\_is\_on = **True**  **while** gam\_is\_on:      scr\_een**.update**()      time**.sleep**(0.1)      #*collision with wall*  **if** (ball**.ycor**() **>=** 280) **or** (ball**.ycor**() **<=** -280):          ball**.bounce\_y**()      #*collision with Paddle*  **if** (ball**.distance**(r\_paddle) **<** 42) **and** (ball**.xcor**() **>=** 330):          ball**.bounce\_x**()  **elif** (ball**.distance**(l\_paddle) **<** 42) **and** (ball**.xcor**() **<=** -330):          ball**.bounce\_x**()      #*elif (ball.xcor() <= -360) or (ball.xcor() >= 360):*      #*gam\_is\_on = False*      ball**.move**()  scr\_een**.exitonclick**()  #*python pong\_main.py* | **class** Ball(turtle**.**Turtle):  **def** **\_\_init\_\_**(self):  **self.**step\_x = 10  **self.**step\_y = 10  **super**()**.\_\_init\_\_**() #*Always use ()*          #*paddle setting*  **self.shape**("circle")  **self.speed**("fastest")  **self.penup**()  **self.color**("#3c412c")  **def** **bounce\_y**(self):  **self.**step\_y \*= -1    **def** **bounce\_x**(self):  **self.**step\_x \*= -1  **def** **move**(self):          new\_x = **self.xcor**() + **self.**step\_x          new\_y = **self.ycor**() + **self.**step\_y  **self.goto**(new\_x, new\_y)  #*python pong\_ball.py* |

**22.8 How to Detect when the Ball goes Out of Bounds**

|  |  |
| --- | --- |
|  |  |
| #*collision with Paddle and out of bound*  **if** (ball**.distance**(r\_paddle) **<** 42) **and** (ball**.xcor**() **>=** 330):          ball**.bounce\_x**()  **elif** (ball**.distance**(l\_paddle) **<** 42) **and** (ball**.xcor**() **<=** -330):          ball**.bounce\_x**()          #*out of bound*  **elif** (ball**.xcor**() **<=** -360) **or** (ball**.xcor**() **>=** 360):          ball**.goto**(0, 0)          ball**.bounce\_x**()  #*python pong\_main.py*    **def** **reset\_position**(self):  **self.goto**(0,0)  **self.bounce\_x**()  #*python pong\_ball.py.py* | **while** gam\_is\_on:      scr\_een**.update**()      time**.sleep**(0.1)      #*collision with wall*  **if** (ball**.ycor**() **>=** 280) **or** (ball**.ycor**() **<=** -280):          ball**.bounce\_y**()      #*collision with Paddle and out of bound*  **if** (ball**.distance**(r\_paddle) **<** 42) **and** (ball**.xcor**() **>=** 330):          ball**.bounce\_x**()  **elif** (ball**.distance**(l\_paddle) **<** 42) **and** (ball**.xcor**() **<=** -330):          ball**.bounce\_x**()      #*Benefit of doing seperately: For updating the score*      #*Detect R paddle misses*  **if** ball**.xcor**() **>=** 360:          ball**.reset\_position**()      #*Detect R paddle misses*  **if** ball**.xcor**() **<=** -360:          ball**.reset\_position**()      ball**.move**()  scr\_een**.exitonclick**()  #*python pong\_main.py* |

**22.9 Score Keeping and Changing the Ball Speed**

Ball speed: practice

sleep\_time = 0.1

gam\_is\_on = **True**

**while** gam\_is\_on:

    scr\_een**.update**()

    time**.sleep**(sleep\_time)

    #*collision with wall*

**if** (ball**.ycor**() **>=** 280) **or** (ball**.ycor**() **<=** -280):

        ball**.bounce\_y**()

    #*collision with Paddle and out of bound*

**if** (ball**.distance**(r\_paddle) **<** 42) **and** (ball**.xcor**() **>=** 330):

        ball**.bounce\_x**()

        #increasing speed *of the ball by reducing*sleep\_time

        sleep\_time /= 1.1

**elif** (ball**.distance**(l\_paddle) **<** 42) **and** (ball**.xcor**() **<=** -330):

        ball**.bounce\_x**()

        #increasing speed *of the ball by reducing*sleep\_time

        sleep\_time /= 1.1

All code at once

**import** turtle

**import** pong\_paddle

**import** pong\_ball

**import** pong\_score

**import** time

scr\_een = turtle**.Screen**()

#*Backgrund color : "#a8c64e" and Turtle color: "#3c412c"*

scr\_een**.title**("PONG")

scr\_een**.bgcolor**("#a8c64e")

scr\_een**.setup**(width= 800, height= 600)

#*animation modification*

scr\_een**.tracer**(0)

#*Paddle setup*

#*Notice Tuple is used*

r\_paddle = pong\_paddle**.Paddle**((350, 0))

l\_paddle = pong\_paddle**.Paddle**((-350, 0))

#*Ball setup. Use tuple for position*

ball = pong\_ball**.Ball**()

#*Score*

score = pong\_score**.Score**()

scr\_een**.listen**()

scr\_een**.onkeypress**(r\_paddle**.**up, "Up")

scr\_een**.onkeypress**(r\_paddle**.**down, "Down")

scr\_een**.onkeypress**(l\_paddle**.**up, "Left")

scr\_een**.onkeypress**(l\_paddle**.**down, "Right")

gam\_is\_on = **True**

**while** gam\_is\_on:

    scr\_een**.update**()

    time**.sleep**(ball**.**ball\_speed)

    #*collision with wall*

**if** (ball**.ycor**() **>=** 280) **or** (ball**.ycor**() **<=** -280):

        ball**.bounce\_y**()

    #*collision with Paddle and out of bound*

**if** (ball**.distance**(r\_paddle) **<** 42) **and** (ball**.xcor**() **>=** 330):

        ball**.bounce\_x**()

**elif** (ball**.distance**(l\_paddle) **<** 42) **and** (ball**.xcor**() **<=** -330):

        ball**.bounce\_x**()

    #*Benefit of doing seperately: For updating the score*

    #*Detect R paddle misses*

**if** ball**.xcor**() **>=** 360:

        ball**.reset\_position**()

        score**.l\_point**()

    #*Detect R paddle misses*

**if** ball**.xcor**() **<=** -360:

        ball**.reset\_position**()

        score**.r\_point**()

    ball**.move**()

scr\_een**.exitonclick**()

#*python pong\_main.py*

**import** turtle

STEP = 20

**class** Paddle(turtle**.**Turtle):

**def** **\_\_init\_\_**(self, posITon):

**super**()**.\_\_init\_\_**() #*Always use ()*

        #*paddle setting*

**self.shape**("square")

**self.shapesize**(stretch\_len=1, stretch\_wid=5)

**self.speed**("fastest")

**self.penup**()

**self.color**("#3c412c")

**self.goto**(posITon)

**def** **change\_pos**(self, step):

        x\_crd = **self.xcor**()

        y\_crd = **self.ycor**()

**self.goto**(x\_crd, y\_crd + step)

**def** **up**(self):

**self.change\_pos**(STEP)

**def** **down**(self):

**self.change\_pos**(-STEP)

#*python pong\_paddle.py*

**import** turtle

**class** Ball(turtle**.**Turtle):

**def** **\_\_init\_\_**(self):

**self.**step\_x = 10

**self.**step\_y = 10

**super**()**.\_\_init\_\_**() #*Always use ()*

        #*paddle setting*

**self.shape**("circle")

**self.speed**("fastest")

**self.penup**()

**self.color**("#3c412c")

**self.**ball\_speed = 0.1

**def** **bounce\_y**(self):

**self.**step\_y \*= -1

**def** **bounce\_x**(self):

**self.**step\_x \*= -1

            #*increasing speed of the ball by reducing sleep-time : "ball.ball\_speed"*

**self.**ball\_speed /= 1.1

**def** **move**(self):

        new\_x = **self.xcor**() + **self.**step\_x

        new\_y = **self.ycor**() + **self.**step\_y

**self.goto**(new\_x, new\_y)

**def** **reset\_position**(self):

**self.goto**(0,0)

**self.bounce\_x**()

**self.**ball\_speed = 0.1

#*python pong\_ball.py.py*

**import** turtle

#*from turtle import Screen*

**class** Score(turtle**.**Turtle):

**def** **\_\_init\_\_**(self):

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

**self.penup**()

**self.hideturtle**()

**self.**l\_score = 0

**self.**r\_score = 0

**self.color**("#3c412c")

**self.show\_score**()

**def** **show\_score**(self):

**self.goto**(-100, 200)

**self.write**(**self.**l\_score, move= **False**, align= "center", font= ("Courier", 80, "normal"))

**self.goto**(100, 200)

**self.write**(**self.**r\_score, move= **False**, align= "center", font= ("Courier", 80, "normal"))

**def** **l\_point**(self):

**self.clear**()

**self.**l\_score += 1

**self.show\_score**()

**def** **r\_point**(self):

**self.clear**()

**self.**r\_score += 1

**self.show\_score**()

#*python pong\_score.py*