Day - 23

**Turtle Crossing: Capstone Project**

Project: Breakdown problem, Start to code

**23.1 Breakdown problem**

* Screen file / main file test: Detect Collision, Score-Control, End of line
* Edit the turtle in player file: Move Turtle
* Edit the car file: Create Cars
* Edit the Score file
* Create the Player Behavior
* Create the Car Behavior
* Detect when the Turtle collides with a Car squish
* Detect when the Player has reached the other side
* Add the Scoreboard and Game Over sequence

**23.2 Create the Player Behavior**

**import** turtle

STARTING\_POSITION = (0, -330)

MOVE\_DISTANCE = 10

FINISH\_LINE\_Y = 330

**class** Player(turtle**.**Turtle):

**def** **\_\_init\_\_**(self):

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

**self.speed**(0)

**self.shape**("turtle")

**self.penup**()

**self.color**("green")

**self.go\_to\_start\_position**()

**self.setheading**(90)

**def** **go\_to\_start\_position**(self):

**self.goto**(STARTING\_POSITION)

**def** **move\_forwrd**(self):

**self.forward**(MOVE\_DISTANCE)

**def** **move\_back**(self):

**self.backward**(MOVE\_DISTANCE)

**def** **is\_at\_finish\_line**(self):

**if** **self.ycor**() **==** FINISH\_LINE\_Y:

**return** **True**

**else**:

**return** **False**

**23.3 Create the Car Behavior**

**import** random

**import** turtle

COLORS = ["red", "orange", "yellow", "green", "blue", "purple"]

STARTING\_MOVE\_DISTANCE = 5

MOVE\_INCREMENT = 10

**class** CarManager():

**def** **\_\_init\_\_**(self):

**self.**all\_cars = []

**self.**car\_speed = STARTING\_MOVE\_DISTANCE

**def** **create\_car**(self):

        #*Reduce the car number by random choice.*

**if** random**.randint**(1, 6) **==** 3:

**self.**car = turtle**.Turtle**(shape= "square")

**self.**car**.penup**()

**self.**car**.color**(random**.choice**(COLORS))

**self.**car**.shapesize**(stretch\_len= 2, stretch\_wid= 1)

            rand\_y = random**.randint**(-300, 300)

**self.**car**.goto**(350, rand\_y)

**self.**all\_cars**.append**(**self.**car)

**def** **move\_cars**(self):

**for** cAr **in** **self.**all\_cars:

            cAr**.backward**(**self.**car\_speed)

**def** **level\_up**(self):

**self.**car\_speed += MOVE\_INCREMENT

#*(0.1 + random.random())*

#*self.shapesize(stretch\_len= 2 + 3\*(0.1 + random.random()) , stretch\_wid= 2\*(0.1 + random.random()))*

**23.4 The Main-file**

**import** time

**import** turtle

**import** turtle\_crossing\_player

**import** turtle\_crossing\_car\_manager

**import** turtle\_crossing\_scoreboard

screen = turtle**.Screen**()

screen**.setup**(width=700, height=700)

screen**.tracer**(0)

plaYer = turtle\_crossing\_player**.Player**()

car\_manage = turtle\_crossing\_car\_manager**.CarManager**()

score = turtle\_crossing\_scoreboard**.Scoreboard**()

screen**.listen**()

screen**.onkeypress**(plaYer**.**move\_forwrd, "Up")

screen**.onkeypress**(plaYer**.**move\_back, "Down")

game\_is\_on = **True**

**while** game\_is\_on:

    time**.sleep**(0.1)

    #*create and move cars*

    car\_manage**.create\_car**()

    car\_manage**.move\_cars**()

    screen**.update**()

    #*Deteet collision with car and turtle*

**for** cAr **in** car\_manage**.**all\_cars:

**if** plaYer**.distance**(cAr) **<** 20:

            game\_is\_on = **False**

            score**.game\_over**()

    #*Detect successfull reach at finish line*

**if** plaYer**.is\_at\_finish\_line**():

        plaYer**.go\_to\_start\_position**()

        car\_manage**.level\_up**()

        score**.increase\_score**()

screen**.exitonclick**()

#*python turtle\_crossing\_main.py*

**23.4 Detect when the Turtle collides with a Car squish**

    #*Deteet collision with car and turtle*

**for** cAr **in** car\_manage**.**all\_cars:

**if** plaYer**.distance**(cAr) **<** 20:

            game\_is\_on = **False**

            score**.game\_over**()

**23.5 Detect when the Player has reached the other side**

    #*Detect successfull reach at finish line*

**if** plaYer**.is\_at\_finish\_line**():

        plaYer**.go\_to\_start\_position**()

        car\_manage**.level\_up**()

        score**.increase\_score**()

**23.6 Add the Scoreboard and Game Over sequence**

**from** turtle **import** Turtle

FONT = ("Courier", 24, "normal")

**class** Scoreboard(Turtle):

**def** **\_\_init\_\_**(self):

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

**self.**score = 1

**self.hideturtle**()

**self.penup**()

**self.goto**(-280, 300)

**self.update\_scoreboard**()

**def** **update\_scoreboard**(self):

**self.write**(f"Level : {**self.**score}", align= "left", font= FONT)

**def** **increase\_score**(self):

**self.**score += 1

**self.clear**()

**self.update\_scoreboard**()

**def** **game\_over**(self):

**self.goto**(0, 0)

**self.write**(f"Game Over", align= "center", font= FONT)