**Step-by-step tutorial by Nuriza and Tamerlan**

**How to create dinosaur game with python and library pygame?**

**What is Pygame?**

Pygame is a third party module designed for creating 2D games with python. It has very simple and intuitive syntax that can allows us to create powerful games quickly. Pygame uses surfaces for drawing. This means whenever we want to draw something to the screen it must be converted to a surface. There are many methods and functions within pygame that can do this easily for us.

**Installing Pygame**

Installing pygame can be as easy as :  
1. Opening your command prompt  
2. Typing *pip install pygame*

However, in many cases this will not work. Below are guides for installing pygame on both windows and mac.

**Windows:**[How to Install Pygame on Windows](https://www.youtube.com/watch?v=AdUZArA-kZw)

**Mac OS:**[How to Install Pygame on Mac](https://www.youtube.com/watch?v=E-WhAS6qzsU)

**Linux:**refer to the Mac OS Guide.

**Creating a Window**

After we import pygame it is a good idea to initialize it, like so:

*import* pygame  
pygame.init()

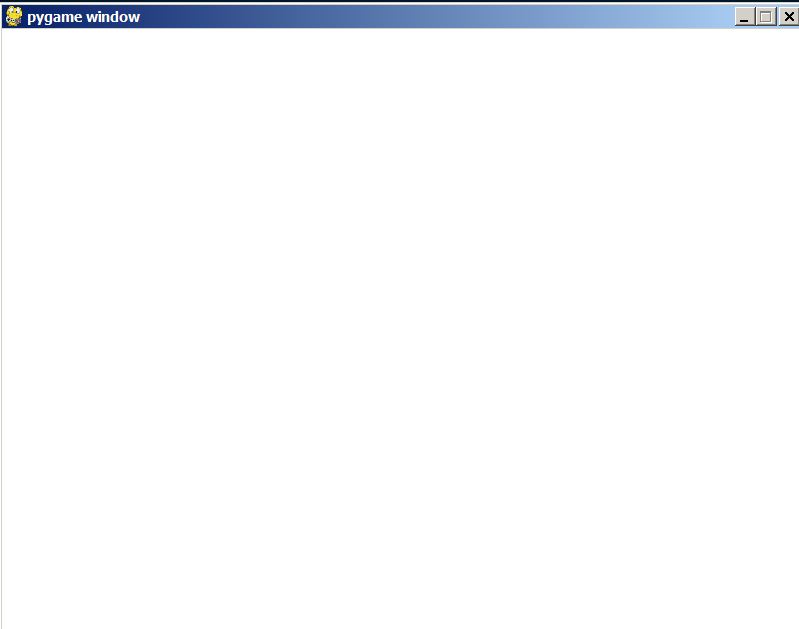
Once we've done that we need to setup display that will represent our game.

*import* pygame  
pygame.init()

display\_width = 800  
display\_height = 600  
  
display = pygame.display.set\_mode((display\_width, display\_height))

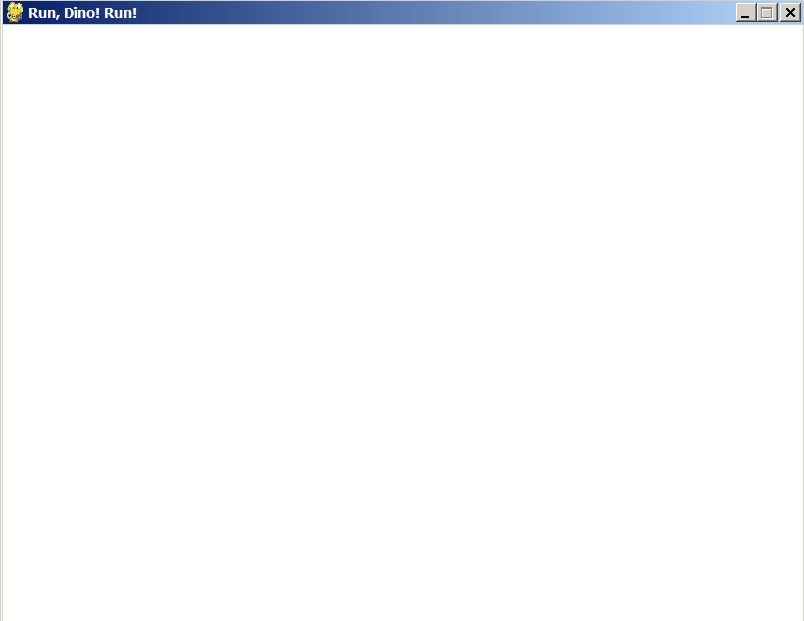
# This code creates a display with width = 800, height = 600

Now when we run the program we get something that looks like this



This is great but we’d like to give our game a more creative name than “pygame window”. To do this we can type the following:

pygame.display.set\_caption("Run, Dino! Run!")



That’s better!

Now we are going to setup our **main-loop or game-loop**. All games have some sort of loop that executes constantly. This loop is responsible for tasks such as checking for events (such as keyboard events or collision), moving objects, updating the display and eventually ending the game.  
In our game we will use a while loop.

Inside the loop we will implement a time delay so we can control the speed of the game. We will also start by checking for some specific events.

*import* pygame  
  
pygame.init()  
  
display\_width = 800  
display\_height = 600  
display = pygame.display.set\_mode((display\_width, display\_height))  
pygame.display.set\_caption('Run, Dino! Run!')  
  
  
*def* run\_game():  
 game = *True  
  
 while* game:  
 *for* event *in* pygame.event.get():  
 *if* event.type == pygame.QUIT:  
 pygame.quit()  
 *quit*()  
  
 display.fill((255, 255, 255))  
 pygame.display.update()  
  
  
run\_game()

Now we can draw a rectangle to the screen to represent our character. We will draw the rectangle in the main loop so that it is constantly redrawn each time we loop.

*import* pygame  
pygame.init()  
display\_width = 800  
display\_height = 600  
display = pygame.display.set\_mode((display\_width, display\_height))  
pygame.display.set\_caption('Run, Dino! Run!')  
x = 50  
y = 50  
width = 40  
height = 60  
vel = 5  
  
pygame.draw.rect(display, (255, 0, 0), (x, y, width, height)) *# This takes: window/surface, color, rect*pygame.display.update() *# This updates the screen so we can see our rectangle  
   
   
def* run\_game():  
 game = *True  
  
 while* game:  
 *for* event *in* pygame.event.get():  
 *if* event.type == pygame.QUIT:  
 pygame.quit()  
 *quit*()  
  
 display.fill((255, 255, 255))  
 pygame.display.update()  
  
  
run\_game()

How to load music and change volume? It is easy! Set volume the max volume is 1.

This music and images must be in one file with pycharm file.

pygame.mixer.music.load("background.mp3")  
pygame.mixer.music.set\_volume(0.4)

You can change icon with this pygame function.

icon = pygame.image.load("icon.jpg")  
pygame.display.set\_icon(icon)

Also you can change size of the image.

health\_image = pygame.image.load("heart.png")  
health\_image = pygame.transform.scale(health\_image, (30, 30))

Here is class object for cactuses.

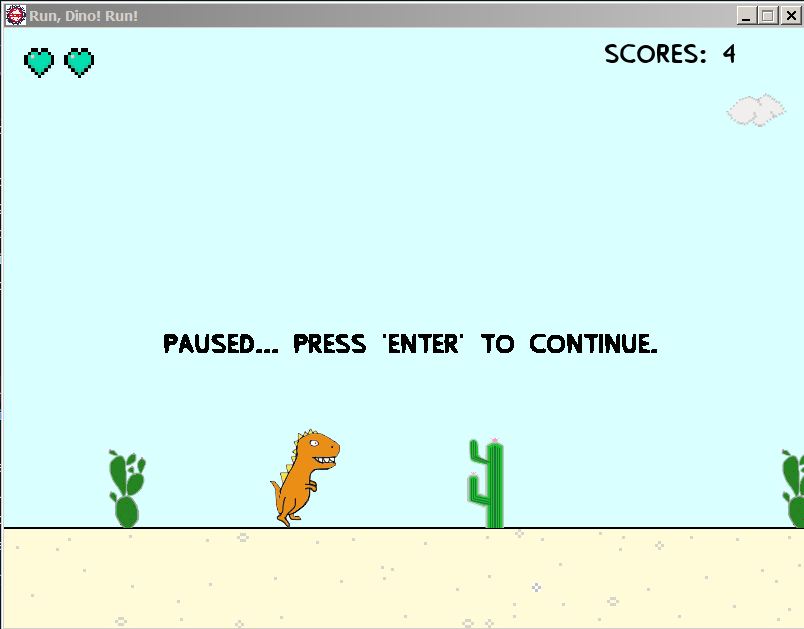
*class* Object:  
 *def \_\_init\_\_*(*self*, x, y, width, image, speed):  
 *self*.x = x  
 *self*.y = y  
 *self*.width = width  
 *self*.image = image  
 *self*.speed = speed  
  
 *def* move(*self*):  
 *if self*.x >= -*self*.width:  
 display.blit(*self*.image, (*self*.x, *self*.y))  
 *self*.x -= *self*.speed  
 *return True  
 else*:  
 *return False  
  
 def* return\_self(*self*, radius, y, width, image):  
 *self*.x = radius  
 *self*.y = y  
 *self*.width = width  
 *self*.image = image  
 display.blit(*self*.image, (*self*.x, *self*.y))

Our dinosaur’s paramaters.

user\_width = 60  
user\_height = 100  
user\_x = display\_width // 3  
user\_y = display\_height-user\_height - 100  
  
cactus\_width = 20  
cactus\_height = 70  
cactus\_x = display\_width - 50  
cactus\_y = display\_height - cactus\_height - 100  
  
clock = pygame.time.Clock()  
make\_jump = *False*jump\_counter = 30  
scores = 0  
max\_scores = 0  
above\_cactus = *False*max\_above = 0

If you know python well and you can read the code, you can easily understand and write your own code! Good luck! Here you can find all my code.

*import* pygame  
*import* random  
pygame.init()  
  
  
display\_width = 800  
display\_height = 600  
  
display = pygame.display.set\_mode((display\_width, display\_height))  
pygame.display.set\_caption("Run, Dino! Run!")  
  
pygame.mixer.music.load("background.mp3")  
pygame.mixer.music.set\_volume(0.4)  
  
jump\_sound = pygame.mixer.Sound("Rrr.wav")  
pygame.mixer.Sound.set\_volume(jump\_sound, 0.4)  
fall\_sound = pygame.mixer.Sound("Bdish.wav")  
pygame.mixer.Sound.set\_volume(fall\_sound, 0.8)  
loss\_sound = pygame.mixer.Sound("loss.wav")  
  
icon = pygame.image.load("icon.jpg")  
pygame.display.set\_icon(icon)  
  
cactus\_img = [pygame.image.load("Cactus0.png"), pygame.image.load  
 ("Cactus1.png"), pygame.image.load("Cactus2.png")]  
cactus\_options = [69, 449, 37, 410, 40, 420]  
stone\_img = [pygame.image.load("Stone0.png"), pygame.image.load("Stone1.png")]  
cloud\_img = [pygame.image.load("Cloud0.png"), pygame.image.load("Cloud1.png")]  
dino\_img = [pygame.image.load("Dino0.png"), pygame.image.load("Dino1.png"),  
 pygame.image.load("Dino2.png"), pygame.image.load("Dino3.png"),  
 pygame.image.load("Dino4.png")]  
  
health\_image = pygame.image.load("heart.png")  
health\_image = pygame.transform.scale(health\_image, (30, 30))  
img\_counter = 0  
health = 5  
  
  
*class* Object:  
 *def \_\_init\_\_*(*self*, x, y, width, image, speed):  
 *self*.x = x  
 *self*.y = y  
 *self*.width = width  
 *self*.image = image  
 *self*.speed = speed  
  
 *def* move(*self*):  
 *if self*.x >= -*self*.width:  
 display.blit(*self*.image, (*self*.x, *self*.y))  
 *self*.x -= *self*.speed  
 *return True  
 else*:  
 *return False  
  
 def* return\_self(*self*, radius, y, width, image):  
 *self*.x = radius  
 *self*.y = y  
 *self*.width = width  
 *self*.image = image  
 display.blit(*self*.image, (*self*.x, *self*.y))  
  
  
user\_width = 60  
user\_height = 100  
user\_x = display\_width // 3  
user\_y = display\_height-user\_height - 100  
  
cactus\_width = 20  
cactus\_height = 70  
cactus\_x = display\_width - 50  
cactus\_y = display\_height - cactus\_height - 100  
  
clock = pygame.time.Clock()  
make\_jump = *False*jump\_counter = 30  
scores = 0  
max\_scores = 0  
above\_cactus = *False*max\_above = 0  
  
  
*def* run\_game():  
 *global* make\_jump  
 pygame.mixer.music.play(-1)  
 game = *True* cactus\_arr = []  
 create\_cactus\_arr(cactus\_arr)  
 land = pygame.image.load("Land.jpg")  
  
 stone, cloud = open\_random\_objects()  
 *while* game:  
 *for* event *in* pygame.event.get():  
 *if* event.type == pygame.QUIT:  
 pygame.quit()  
 *quit*()  
 keys = pygame.key.get\_pressed()  
 *if* keys[pygame.K\_SPACE]:  
 make\_jump = *True  
 if* keys[pygame.K\_ESCAPE]:  
 pause()  
 *if* keys[pygame.K\_UP]:  
 pygame.mixer.music.stop()  
 *if* keys[pygame.K\_DOWN]:  
 pygame.mixer.music.play()  
 *if* make\_jump:  
 jump()  
 count\_scores(cactus\_arr)  
  
 display.blit(land, (0, 0))  
 print\_text(" Made by Nuriza and Tamerlan, "  
 "students of Ala-Too ", 30, 150)  
 print\_text('Score: ' + *str*(scores), 600, 10)  
  
 draw\_array(cactus\_arr)  
 move\_objects(stone, cloud)  
 draw\_dino()  
 *if* check\_collision(cactus\_arr):  
 pygame.mixer.music.stop()  
 pygame.mixer.Sound.play(fall\_sound)  
 *# if not check\_health():* game = *False* show\_health()  
  
 pygame.display.update()  
 clock.tick(80)  
 *return* game\_over()  
  
  
*def* jump():  
 *global* user\_y, jump\_counter, make\_jump  
 *if* jump\_counter >= -30:  
 *if* jump\_counter == 30:  
 pygame.mixer.Sound.play(jump\_sound)  
 *elif* jump\_counter == -26:  
 pygame.mixer.Sound.play(fall\_sound)  
  
 user\_y -= jump\_counter / 2.5  
 jump\_counter -= 1  
 *else*:  
 jump\_counter = 30  
 make\_jump = *False  
  
  
def* create\_cactus\_arr(array):  
 choice = random.randrange(0, 3)  
 img = cactus\_img[choice]  
 width = cactus\_options[choice\*2]  
 height = cactus\_options[choice\*2 + 1]  
 array.append(Object(display\_width + 20, height, width, img, 4))  
  
 choice = random.randrange(0, 3)  
 img = cactus\_img[choice]  
 width = cactus\_options[choice \* 2]  
 height = cactus\_options[choice \* 2 + 1]  
 array.append(Object(display\_width + 300, height, width, img, 4))  
  
 choice = random.randrange(0, 3)  
 img = cactus\_img[choice]  
 width = cactus\_options[choice \* 2]  
 height = cactus\_options[choice \* 2 + 1]  
 array.append(Object(display\_width + 600, height, width, img, 4))  
  
  
*def* find\_radius(array):  
 maximum = *max*(array[0].x, array[1].x, array[2].x)  
 *if* maximum < display\_width:  
 radius = display\_width  
 *if* radius - maximum < 50:  
 radius += 280  
 *else*:  
 radius = maximum  
  
 choice = random.randrange(0, 5)  
 *if* choice == 0:  
 radius += random.randrange(10, 15)  
 *else*:  
 radius += random.randrange(250, 400)  
 *return* radius  
  
  
*def* object\_return(objects, obj):  
 radius = find\_radius(objects)  
  
 choice = random.randrange(0, 3)  
 img = cactus\_img[choice]  
 width = cactus\_options[choice \* 2]  
 height = cactus\_options[choice \* 2 + 1]  
 obj.return\_self(radius, height, width, img)  
  
  
*def* draw\_array(array):  
 *for* cactus *in* array:  
 check = cactus.move()  
 *if not* check:  
 object\_return(array, cactus)  
  
  
*def* open\_random\_objects():  
 choice = random.randrange(0, 2)  
 img\_of\_stone = stone\_img[choice]  
  
 choice = random.randrange(0, 2)  
 img\_of\_cloud = cloud\_img[choice]  
 stone = Object(display\_width, display\_height - 80, 10, img\_of\_stone, 4)  
 cloud = Object(display\_width, 80, 70, img\_of\_cloud, 2)  
 *return* stone, cloud  
  
  
*def* move\_objects(stone, cloud):  
 check = stone.move()  
 *if not* check:  
 choice = random.randrange(0, 2)  
 img\_of\_stone = stone\_img[choice]  
 stone.return\_self(display\_width, 500 + random.randrange(10, 80),  
 stone.width, img\_of\_stone)  
  
 check = cloud.move()  
 *if not* check:  
 choice = random.randrange(0, 2)  
 img\_of\_cloud = cloud\_img[choice]  
 cloud.return\_self(display\_width, random.randrange(10, 200),  
 cloud.width, img\_of\_cloud)  
  
  
*def* draw\_dino():  
 *global* img\_counter  
 *if* img\_counter == 25:  
 img\_counter = 0  
 display.blit(dino\_img[img\_counter // 5], (user\_x, user\_y))  
 img\_counter += 1  
  
  
*def* print\_text(message, x, y, font\_color=(0, 0, 0),  
 font\_type="PingPong.ttf", font\_size=30):  
 font\_type = pygame.font.Font(font\_type, font\_size)  
 text = font\_type.render(message, *True*, font\_color)  
 display.blit(text, (x, y))  
  
  
*def* pause():  
 paused = *True* pygame.mixer.music.pause()  
 *while* paused:  
 *for* event *in* pygame.event.get():  
 *if* event.type == pygame.QUIT:  
 pygame.quit()  
 *quit*()  
 print\_text("Paused... Press 'Enter' to continue. ", 160, 300)  
 keys = pygame.key.get\_pressed()  
 *if* keys[pygame.K\_RETURN]:  
 paused = *False* pygame.display.update()  
 clock.tick(15)  
 pygame.mixer.music.unpause()  
  
  
*def* check\_collision(barriers):  
 *for* barrier *in* barriers:  
 *if* barrier.y == 449: *# Little cactus  
 if not* make\_jump:  
 *if* barrier.x <= user\_x + user\_width - 30 <= \  
 barrier.x + barrier.width:  
 *if* check\_health():  
 object\_return(barriers, barrier)  
 *return False  
 else*:  
 *return True  
 elif* jump\_counter >= 0:  
 *if* user\_y + user\_height - 5 >= barrier.y:  
 *if* barrier.x <= user\_x + user\_width - 35 <= \  
 barrier.x + barrier.width:  
 *if* check\_health():  
 object\_return(barriers, barrier)  
 *return False  
 else*:  
 *return True  
 else*:  
 *if* user\_y + user\_height - 10 >= barrier.y:  
 *if* barrier.x <= user\_x <= barrier.x + barrier.width:  
 *if* check\_health():  
 object\_return(barriers, barrier)  
 *return False  
 else*:  
 *return True  
 else*:  
 *if not* make\_jump:  
 *if* barrier.x <= user\_x + user\_width - 5 <= \  
 barrier.x + barrier.width:  
 *if* check\_health():  
 object\_return(barriers, barrier)  
 *return False  
 else*:  
 *return True  
  
 elif* jump\_counter == 10:  
 *if* user\_y + user\_height - 5 >= barrier.y:  
 *if* barrier.x <= user\_x + user\_width - 5 <= \  
 barrier.x + barrier.width:  
 *if* check\_health():  
 object\_return(barriers, barrier)  
 *return False  
 else*:  
 *return True  
 elif* jump\_counter >= -1:  
 *if* user\_y + user\_height - 5 >= barrier.y:  
 *if* barrier.x <= user\_x + user\_width - 35 <=\  
 barrier.x + barrier.width:  
 *if* check\_health():  
 object\_return(barriers, barrier)  
 *return False  
 else*:  
 *return True  
 else*:  
 *if* user\_y + user\_height - 10 >= barrier.y:  
 *if* barrier.x <= user\_x + 5 <= barrier.x + barrier.width:  
 *if* check\_health():  
 object\_return(barriers, barrier)  
 *return False  
 else*:  
 *return True  
 return False  
  
  
def* count\_scores(barriers):  
 *global* scores, max\_above  
 above\_cactus = 0  
 *if* -20 <= jump\_counter < 25:  
 *for* barrier *in* barriers:  
 *if* user\_y + user\_height - 5 <= barrier.y:  
 *if* barrier.x <= user\_x <= barrier.x + barrier.width:  
 above\_cactus += 1  
 *elif* barrier.x <= user\_x + user\_width <= \  
 barrier.x + barrier.width:  
 above\_cactus += 1  
 max\_above = *max*(max\_above, above\_cactus)  
  
 *else*:  
 *if* jump\_counter == -30:  
 scores += max\_above  
 max\_above = 0  
  
  
*def* game\_over():  
 *global* scores, max\_scores  
 *if* scores > max\_scores:  
 max\_scores = scores  
 stopped = *True  
 while* stopped:  
 *for* event *in* pygame.event.get():  
 *if* event.type == pygame.QUIT:  
 pygame.quit()  
 *quit*()  
 print\_text("Game over... Press 'Enter' to play again,"  
 " 'Esc' to exit ", 22, 300)  
 print\_text('Max score: ' + *str*(max\_scores), 300, 350)  
 keys = pygame.key.get\_pressed()  
 *if* keys[pygame.K\_RETURN]:  
 *return True  
 if* keys[pygame.K\_ESCAPE]:  
 *return False* pygame.display.update()  
 clock.tick(15)  
  
  
*def* show\_health():  
 *global* health  
 show = 0  
 x = 20  
 *while* show != health:  
 display.blit(health\_image, (x, 20))  
 x += 40  
 show += 1  
  
  
*def* check\_health():  
 *global* health  
 health -= 1  
 *if* health == 0:  
 pygame.mixer.Sound.play(loss\_sound)  
 *return False  
 else*:  
 pygame.mixer.Sound.play(fall\_sound)  
 *return True  
while* run\_game():  
 scores = 0  
 make\_jump = *False* jump\_counter = 30  
 user\_y = display\_height - user\_height - 100  
 health = 5  
pygame.quit()  
*quit*()



Pygame documentation: <http://www.pygame.org/docs/>

Python documentation: <https://www.python.org/doc/>

Python: [https://www.python.org/](https://www.youtube.com/redirect?redir_token=QUFFLUhqa3Z0MVBZZjlqT2s3MkxLQXRUa2hGeHZzY0N0QXxBQ3Jtc0trQnF2Y0RiZ2xQS0JHU01XMDlpcktXalREZEZMcDBJcE1XMWVtbzdMc0hZVHdvUWtaa0JzTkhhemJhQTBTRmNnYWxXNFdKNnZOcTF6OXk0ekNFS2tuNi1zUDI3Z2hQbFh3YnhMNjFPOG9yN1RPbDE4MA%3D%3D&q=https%3A%2F%2Fwww.python.org%2F&event=video_description&v=ge_1FK_E-fs)

Pycharm: [https://www.jetbrains.com/](https://www.youtube.com/redirect?redir_token=QUFFLUhqbVZfRnpvbTVTVDhPYmdPMVg5Yl9IZmctSm9NZ3xBQ3Jtc0trb3MwNEtsdVZ6YWlhX3ZWQi02SExVNWRZQVcwTEZvREUyMGNuN0dyYkdYc0I3M2VTZEpsM0dUcnNMNVRTY0VnZ2ppaGMxem1yemh2cGhDRm9WUmZzN3YwR3pPbVZMSDB0azRsd0JCTUx1MzNWV3Jpcw%3D%3D&q=https%3A%2F%2Fwww.jetbrains.com%2F&event=video_description&v=ge_1FK_E-fs)

GitHub page: <https://github.com/PaishankyzyNuriza/Dino>

Whatsapp: 0553883155

Gmail: [paishanovanuriza@gmail.com](mailto:paishanovanuriza@gmail.com)