



AP Computer Science Principles Python



Unit 8: Game Programming
in Python
U8L1: Intro to Pygame

February
rdas3@schools.nyc.gov

Do Now!

We are (mostly) done with replit! Follow these directions

1. Open up PyCharm on your desktop. You may need to click "Agree" and go through a few other dialogues. After that, you should see an open to create a **New Project**
2. Name your project "U8L1-Intro to Pygame-YOURNAME"
(First Initial, Last Name) *as long as it's unique!*

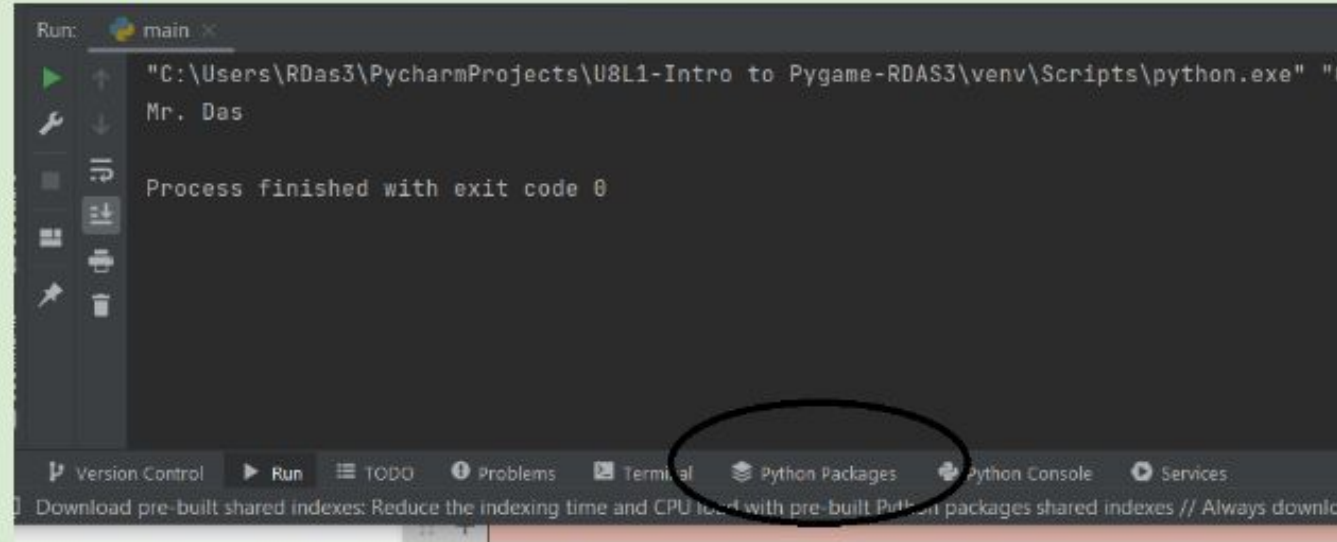
Location: C:\Users\RDas3\PycharmProject\U8L1-Intro to Pygame - RDAS3

3. Your **main.py** file will have a bunch of stuff in it. Delete it and replace it with a simple print statement to print your name.
4. Run your program to make sure everything works! At the top right, you should see a **Green Arrow**. Click it to run your program.



Installing pygame!

For each project, you'll have to install **Pygame**. Go to the bottom of the screen and click on **Python Packages**.



Click that, then type "pygame" in the search field. Once it shows up, click the **Install Package** button on the right.



Now you're ready to go!!! **Remember** that all of your code is only on the school computer. Make sure to **save** your code (you can either do it on replit or just create **Google Doc** with your code in it).

Demo! Setting up a Window with Text

Slides with all the details are included. Step by step instructions are also in today's document!

DEMO

Step 1: Starter Code and Basic Set Up!

```
import pygame

# set up pygame modules
pygame.init()
pygame.font.init()
font = pygame.font.SysFont('Arial', 15)
pygame.display.set_caption("AP CSP Pygame!")

# set up variables for the display
size = (400, 300)
screen = pygame.display.set_mode(size)
```

This will be your starter code for most **pygame** projects. First we import pygame, initialize some modules for future use, create a **font** object to write text, and set the caption of the window!

Lastly, you specify the size of the window and create a **screen** object that we will use later as well.

DEMO

Step 2: Setting up Text (Rectangles galore!)

```
display_name = font.render("Mr. Das", True, (255, 255, 255))
```

Everything (except the window background) that is displayed on the screen is a **Rectangle object**

For text, the `render` method creates a **Rectangle** object for us using the font we created earlier.

The `render` method takes three arguments:

1. The `String` you want to render (in this case, "Mr. Das")
2. A `boolean` value that allows you to smooth out the font so it looks nicer (we should really always set this to `True`).
3. A `tuple` that represents the RGB value for the text (in this case, we are making the white).

Step 3: Main Program Loop

```
# ----- Main Program Loop -----  
while run:  
    # --- Main event loop  
    for event in pygame.event.get(): # User did something  
        if event.type == pygame.QUIT: # If user clicked close  
            run = False  
  
    screen.fill((r, g, b))  
    screen.blit(display_name, (0, 0))  
    screen.blit(display_slogan, (150, 120))  
    screen.blit(display_date, (270, 280))  
    pygame.display.update()  
  
# Once we have exited the main program loop we can stop the game engine:  
pygame.quit()
```

Every **pygame** program will have a **main loop** and what is called your "event loop". The **main loop** will stop when the user closes the window*. We will talk more about the event loop in a later lesson. Copy and paste this code into your program.

****NOTE:** In later programs, you may choose to stop the main loop in other cases (when the player wins, loses, or if you want to show a different menu!)

DEMO

Understanding screen.fill and screen.blit

`screen` is created earlier in the program and refers to the main window of our program.

```
screen.fill((0, 0, 0))
```

The `fill` function allows us to set a background color for the window (here it's black).

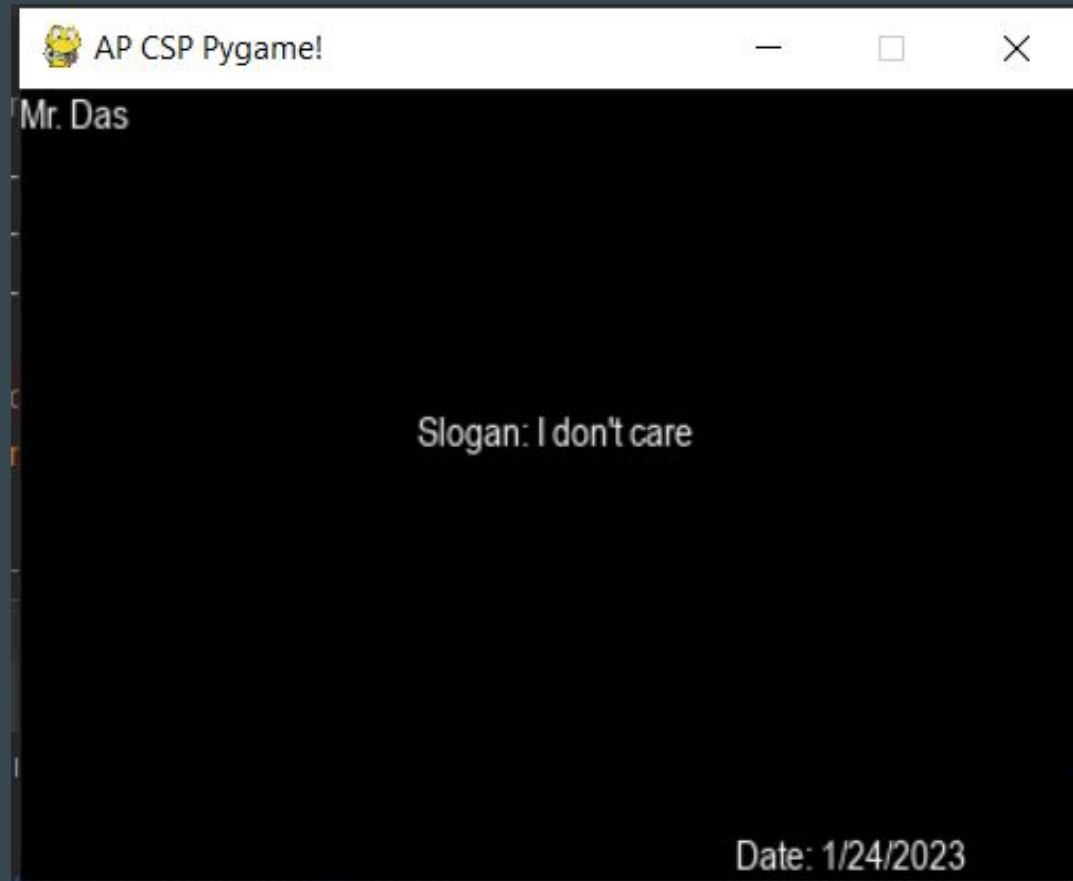
```
screen.blit(display_name, (0, 0))
```

The `blit` function allows us to place **Rectangles** that we've previously rendered onto our main surface. Here, we are rendering `display_name` onto the position 0,0 (the top left of the window is 0, 0).

`pygame.display.update()` does a FULL refresh of the screen based on what you placed on it with `fill` and `blit`.

Your task!

Add two more **text rectangles** to your window and place one in the middle (your slogan), and one on the bottom right (the date).



Visualizing the x,y plane on a window

