Pygame

# Task 1 – there’s your template lets code

Learn it

Pygame is a cross-platform set of Python modules designed for creating 2D games. The modules are designed to be simple, easy to use, and fun—a key part of Pygame's ideology.

Research it

<http://www.pygame.org/>

Pygame was first released in 2000 and has a large community of developer. Have a look at the “awesome” section of the web site.

Template

Pygame requires several line of code to get things up and running, this has been provided for you in the form of a template to use of all your pygame projects.

Create a folder called pygameIntro.

Create a new project using pycharm called pygameCode in the pygameIntro folder.

Download the template\_pygame.py to your pygameIntro folder

// this bit will need trying on students computers….

The template has 2 main sections:

* Game Setup – runs once when the game loads
* Game loop – runs again and again (loops) 60 times a second



Game loop

Setup

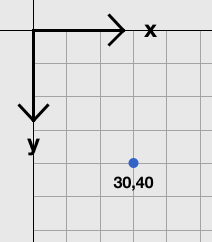
Try it:

Run the template it should create a new window that is black, 1000 \* 800 pixels and closes when you click on the new windows X – that means it’s working!

Learn it

Screen geometry

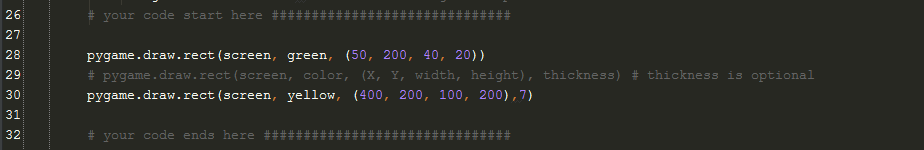
Unlike Scratch, Pygame and most professional game engines have the origin x = 0, y = 0 at the top left hand corner.



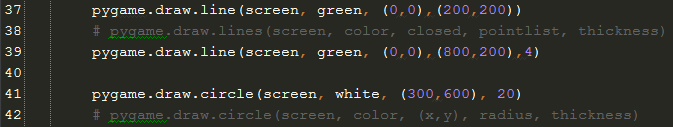
Try it

Pygame has several build in shapes:

Try a rectangle first



Now try creating a few different shapes by inserting the code above into your game loop.



Badge it:

Silver

* Draw and filled and unfilled example of the 3 basic shapes of different colors and in different locations.

<http://www.rapidtables.com/web/color/RGB_Color.htm>

Gold

* Add comments below your code explaining, in your own words, how RGB colors work.

Platinum

* Create some variables for some new colors and use them when creating shapes

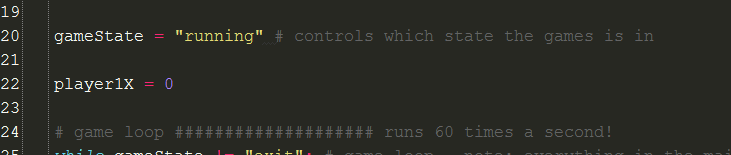
**# Upload your python file #**

Task 2 - Move it! (can be continuation or can be started from fresh file)

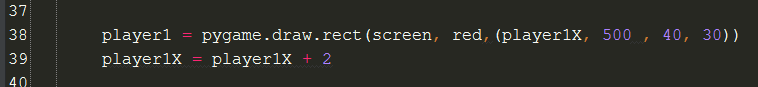
We are going to create another shape but in a slightly different way and make it more around.

Try it

1. In the game setup create a variable player1X and assign it the value of 0

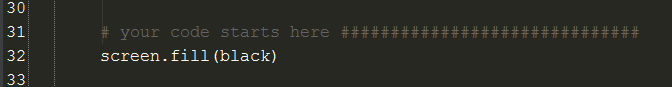


1. Draw player1 in the game loop and increment the X position by 2 pixels every time the game loop runs.



Did this work as expected? Discuss what has happened.

Learn it



Fill the whole screen with a color, this removes any previous drawings.

Badge it

Platinum

Silver

* Add comments to your code and upload it.

Gold

* Add another shape and make it move top to bottom

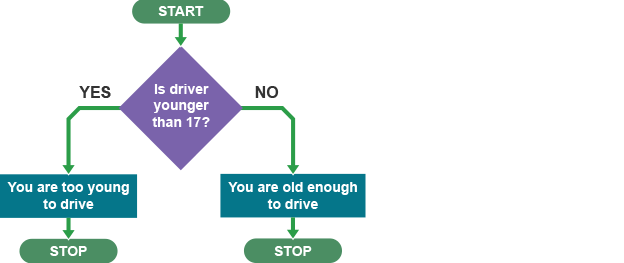
Platinum

Add another shape and make it move diagonally. Try the arc and polygon shapes

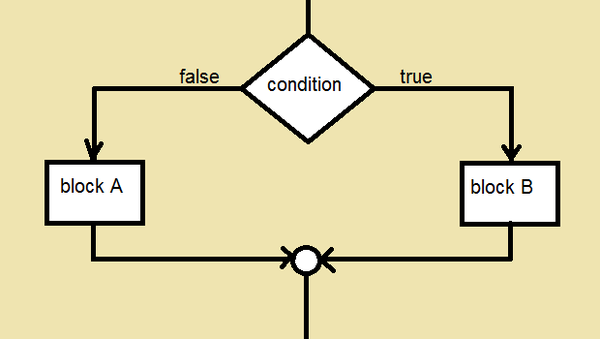
Selection

Learn it

Selection one of the three basic logic structures in computer programming. The other two logic structures are sequence and iteration. In a selection structure, a question is asked, and depending on the answer, the program takes one of two courses of action, after which the program moves on to the next event.



// fix image t/f wrong way

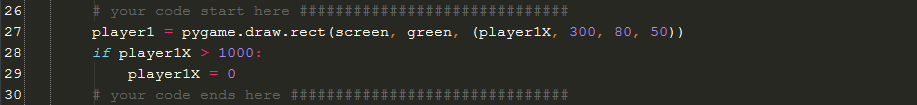


We are going to use value the player1X variable to determine if player1 has moved off the screen.

First we need to create a variable to hold player1’s position



Then check if he has gone over the edge of the screen



Badge it

Silver

* Add comments to all your code

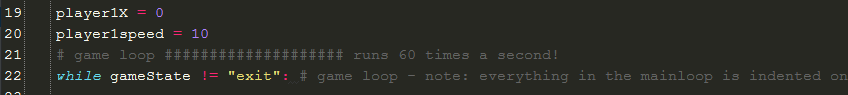
Gold

# Create a circle that:

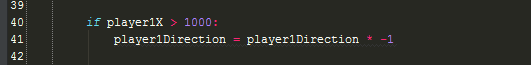
* appears at x= 100, y = 100
* moves to x = 100, y = 700
* reappears at 100, 100

Instead of making the player reappear lets make him bounce.

We will need to create a variable to hold the speed and direction of the player – this must be done before the game loop



We will also need to change the if statement



Try it

The value of player1Direction all also determine his speed

Platinum

* Add another if statement to change his direction when his hits the other side of the screen.

Extension

* Make the player move diagonally and bounce of all side of the screen

# Task 3 Interrupt Events and how to handle them

All games have interaction with the user, in computing these are known as **Interrupt Events.**

The code that checks if the user has done something is called an **Event Listener**

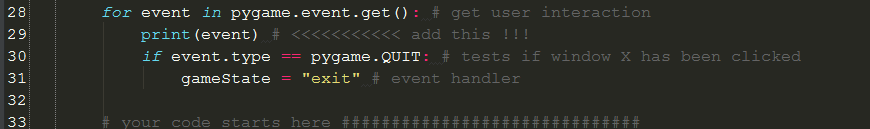
When a specific event is head by the Event Lister

An action is then performed by the **Event Handler** code**.**

If we look carefully at the template you will see that the event of the user closing the window is already handled.

Try it:

Add the print(event) statement



Take a look at the interpreter window while moving the mouse and pressing keys.

Try pressing the ALT key, the returned events are mod:

<Event(2-KeyDown {'mod': 0, 'unicode': '', 'scancode': 56, 'key': 308})>

<Event(3-KeyUp {'mod': 0, 'key': 308, 'scancode': 56})>

NOTE - key up and key down are different events

Badge it

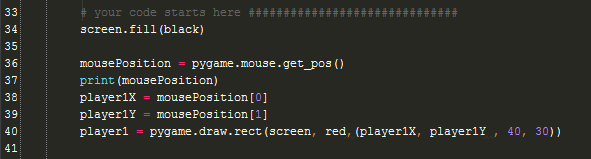
Silver

# Add comments to the bottom the your python code that are the events returned for the following:

* Right mouse button
* Mouse wheel press
* Mouse wheel spin
* Left mouse button
* Pressing the X to close the window
* The up arrow is pressed

Mouse Events

Lets make player1 follow the mouse!



Badge it:

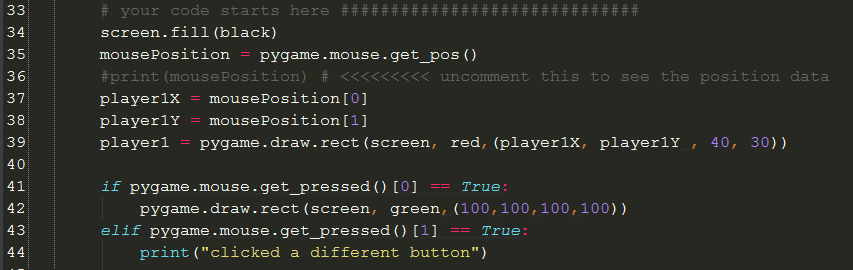
Gold

* In comments at the bottom of your code:
* Why is the mouse cursor NOT in the center of the player?

Platinum

* Fix this bug.
* Add comments to all your code above

# Task 4 - Mouse Clicks



Badge

Silver

# Add comments to the code above

Gold and Platinum

* Modify the code so a different shape is draw depending on which mouse button is pressed.
* The badge level will be determined by your level of creativity
* No badge will be if your game does not run when handed in

# Task 5 Keyboard Events

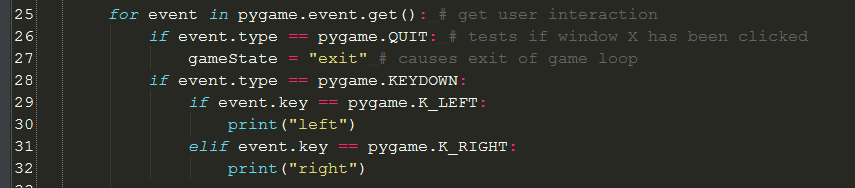
Open your pygame template .py file

As we are making a game for the PC platform the keyboard is a very important input tool.

We are going to control the moment of a player on the screen with the arrow keys

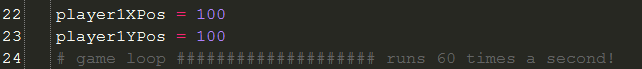
Try it:

Add some addition code to your event listener

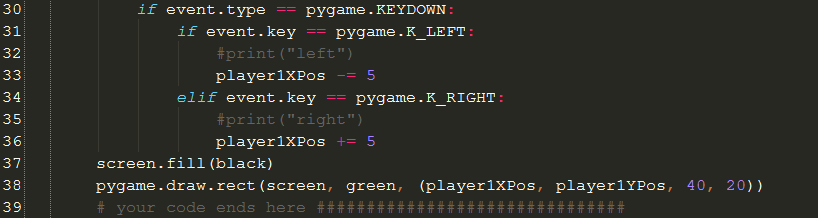


Note – the level of indentation is critical and easy to mess up!

Add 2 variable outside of your game loop



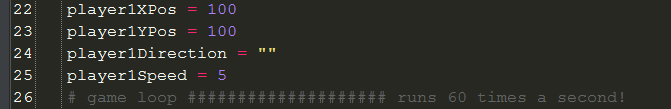
Add the player to the screen:



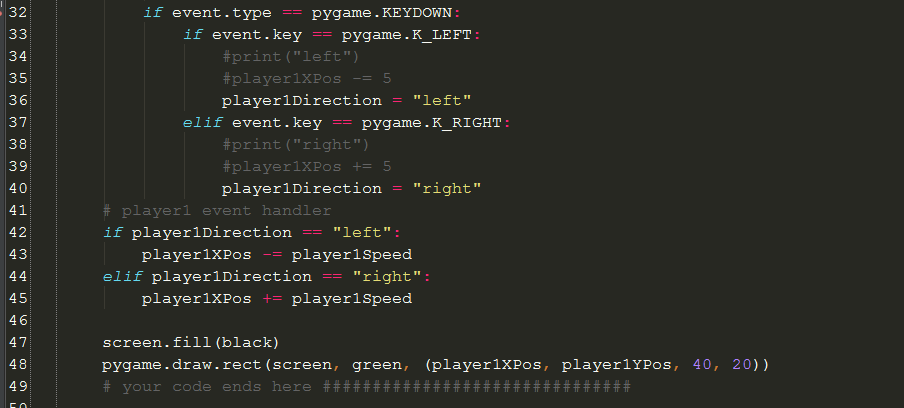
You will find the player moves each time the key is press. This is good for some games but other it is desirable to keep the player moving until another key is pressed. Currently the Event Listener and the Event Handler are in the same block of code.

We will separate the two:

We will need 2 new variable outside of the game loop:



And modify:



Badge it

Silver

# Add comments to your code

Gold

# Add the functionality to move the player up and down

Platinum

# Add the functionality to increase and decrease player1Speed with the Q and W keys

// maybe extent to complete basic pong?

Extensions

# You will notice the player1Speed can go below 0 fix this bug

# Add a feature to make player1 invisible when the space bar is pressed

# Task 6 - Images

Open your pygame template .py file and save a copy as PygameImages.py

Basic shapes are ok but images are much more fun.

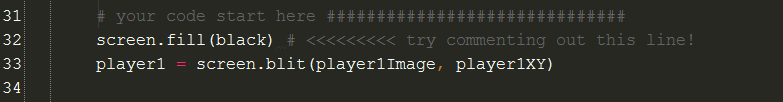
Try it:

First we must load the image ready for later use:

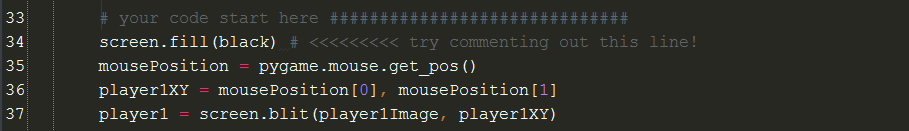
Note – don’t do this in the game loop, loading the image 60 times a second will not lead to good game performance.



Then display the image on the screen:



We are now going to make the sprite follow the mouse – we’ve done this before but we are going to improve our method.



Create in comment at the bottom of your code a bug list:

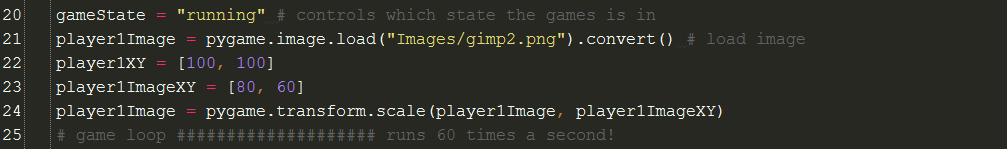
# ## Bug List

* Mouse curser not centred on image
* Mouse curser visible – looks ugly
* Cannot change image size

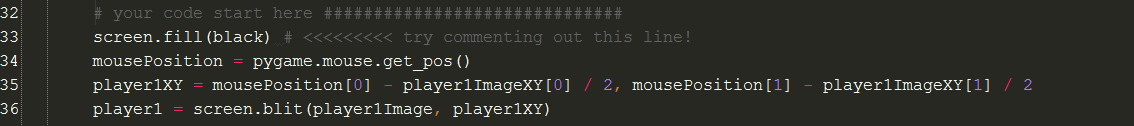
To centre the mouse pointer we need to know the size of the image but we will be changing the image to a different size in the future.

Try it:

To solve this we will set the image size

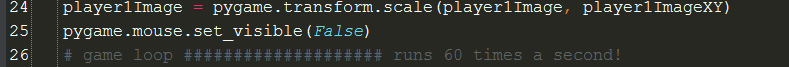


Try it:



Now fix the mouse pointer

The mouse pointer look



Badge it

Silver

# add comments to you code explaining each new line especially line 35

Gold

# Below your code add comment explaining what gimp2 is

# load a different image and correct the player1ImageXY values

Platinum

# Use Gimp2 to cut out a sprite from one for the sprite sheets

# add it as a second player

Extensions

Animate a player by using an event to change the image

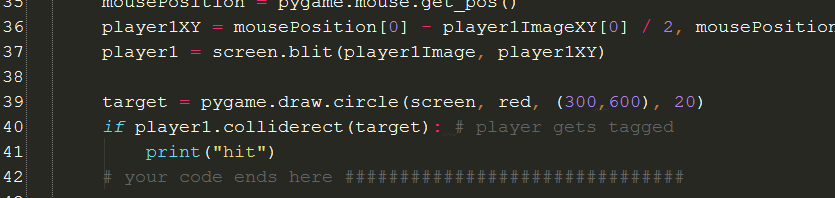
# Task 7- Collisions

// continuation of Images

Open you PygameImages.py file

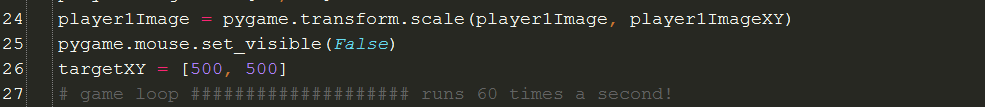
We will create another sprite using a basic shape called target

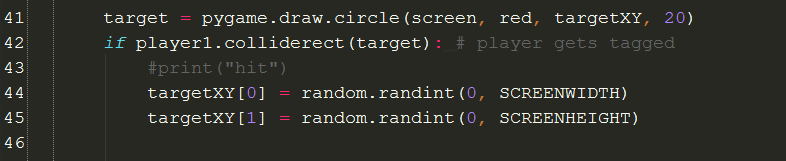
Then detect if player1 has collided with the target



We need to be able to move the target

Create a variable targetXY outside of the game loop





Badge it:

Silver

Add comments to your code

Gold

Create a variable called score and print it every time player1 collides with target

Platinum

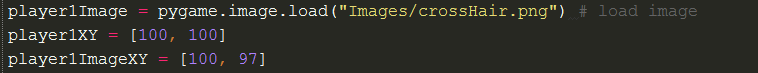
Change the target into an image

Extension

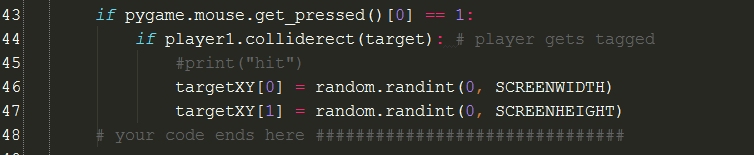
End the game if a score of 10 is reached

Task 7 – Click to shoot continuation of task 6

We are going to all a change the player1 sprite image



Currently the target is hit when it is touched by the player. We can change this to a combination of a collision and a mouse click.



Badge it

Silver

* At the bottom of your code add comments in your own world describing what an image alpha channel is and how it is useful in games

Gold

The game has a bug the target is draw on top of the crosshair.

Fix this bug and explain in the comment why it occurred

Platinum

The game has a bug the fire button can be held down.

# Task 8 – Text me

Text

Open your pygame template and save a copy as pygame\_textFuction.py

Almost all games require text to be displayed on the screen because of this it would be good to make something we can use again and again without writing the code again and again.

In computing this is called a function. It is an independent block of code who lives outside the main program and only runs when asked to do so by the main program.

Function always sit at the top of the code after the import statements.

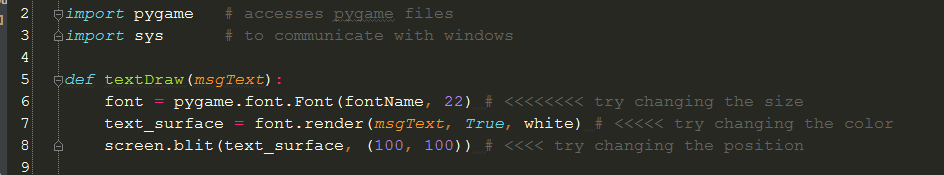
Try it:

First we need to load a font from the PCs font collection and store it in a variable for later us.

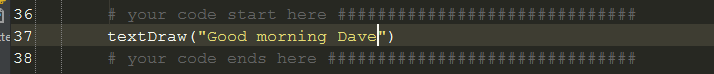


Build the function

At the top of your code under the import statements build you function

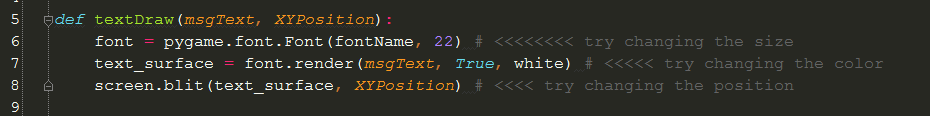


We now need to tell the function that we’d like it to run and what it should draw on the screen. We do this from the main body of your code.

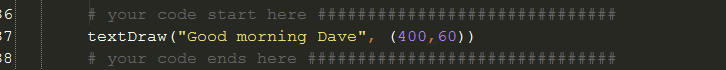


It would be much more useful if we could tell the textDraw function the X, Y coordinates we’d like the text to go.

Add another parameter to your function



Then modify your function call



Badge

Silver

* Add another argument to the function so you can send the size of text you want to the function
* Add another 5 function call to product text message in different locations and sizes.

Gold

Change the font of the text

Extension

Use the function to display a mouse click counter

Add another argument for text color

Platinum

