Chapter 2: Racecar

Description:

Today we will be making cars move around on your screen. The idea is that we want to make pictures clickable and moveable across the screen. To do this we will be using something called pygame.

Why?: This chapter will give you a chance to use an imported library as well as experience new ways to get user input.

Resources:

- Python 3
- A Code editor

To-do

- Display cars
- Let users move cars
- Crash into walls
- Allow users to select cars

Step 1: Initialize the pygame window

```
import pygame

window_x = 1000
window_y = 1000

pygame.init()

gameDisplay = pygame.display.set_mode(window_x,window_y)
pygame.display.set_caption('Race')

pygame.quit()
quit()
```

* On the first line you may notice we are saying import, this means we are grabbing pre-written code (package). This is code that another developer has made and lets us use to make our lives easier.

* This code is the basic code to create a display window. The size of the window and the instructions to terminate it.

Import pygame

Set x and y sizes to 1000

Initiate pygame
Set game window sizes from x and y variables
Set the title to the window to race
Exit pygame

```
import pygame

window_x = 1000
window_y = 1000
black = (0,0,0)

x =  (window_x/2)
y = (window_y/2)
x_change = 0
car_speed = 0

race_end = False

pygame.init()

gameDisplay = pygame.display.set_mode((window_x,window_y))
pygame.display.set_caption('Race')
racecar = pygame.image.load('images/racecar.png')

def car(x,y):
    gameDisplay.blit(racecar, (x,y))

while not race_end:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            race_end = True

    gameDisplay.fill(black)
    car(x,y)

    pygame.display.update()

pygame.quit()
quit()
```

* pygame.image.load does exactly what it sounds like. It will load the image at that location. What isn't said is that without specifying it will look in your current folder.

* The while loop that we have added now is the instructions of what to do while the game is on, The instructions now say to continue the game until we quit the window (pygame. QUIT) which will change the control variable to end the while loop.

* Additionally we will display the racecar image with the car function which will show the car at the x and y coordinates. pygame.display.update() would then update the display.

Import pygame

Set x and y sizes to 1000
Set a variable called black to a tuple to represent the color in RGB
Set two variables x and y with half of the values from the window x and y values. (Middle point of the window)
Set x_change and car_speed to 0 for setting the new x position and simulating speed
Set race_end to signal the end of the program
Initiate pygame
Set game window sizes from x and y variables
Set the title to the window to race
Create a variable to hold an image of a race car
Define function car
    Update image to a new location
While race hasn't ended
    For every event detected by pygame
        If the event is quit
            End race
    Fill background as color stored in the black variable
    Call car function
    Reset window display
Exit pygame

```
import pygame

window_x = 1000
window_y = 1000
black = (0,0,0)

x =  (window_x/2)
y = (window_y/2)
x_change = 0
```

```
y_change = 0
car_speed = 0

race_end = False

pygame.init()

gameDisplay = pygame.display.set_mode((window_x,window_y))
pygame.display.set_caption('Race')
racecar = pygame.image.load('images/racecar.png')

def car(x,y):
    gameDisplay.blit(racecar, (x,y))

while not race_end:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            race_end = True

    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_LEFT:
            x_change = -5
        elif event.key == pygame.K_RIGHT:
            x_change = 5
        elif event.key == pygame.K_UP:
            y_change = -5
        elif event.key == pygame.K_DOWN:
            y_change = 5
    if event.type == pygame.KEYUP:
        if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
            x_change = 0
        if event.key == pygame.K_UP or event.key == pygame.K_DOWN:
            y_change = 0

    x += x_change
    y += y_change
    gameDisplay.fill(black)
    car(x,y)

    pygame.display.update()
```

```
pygame.quit()
quit()
```

* x and y coordinates are where the car will be located on the pygame window.

* When we are waiting for event.key we are using an event listener. This is a function that waits for the user to put in a specific action. In our case, a key is being pressed.

* We then update x and y inputs to show it on display

Import pygame

Set x and y sizes to 1000
Set a variable called black to a tuple to represent the color in RGB
Set two variables x and y with half of the values from the window x and y values. (Middle point of the window)
Set x_change and car_speed to 0 for setting the new x position and simulating speed
Set y_change for the new y position
Set race_end to signal the end of the program
Initiate pygame
Set game window sizes from x and y variables
Set the title to the window to race
Create a variable to hold an image of a race car
Define function car
    Update image to a new location
While race hasn't ended
    For every event detected by pygame
        If the event is quit
            End race
        If the event is up, down, right, and left arrows then add or subtract 5 from y or x coordinates
    Update x and y coordinate
    Fill background as color stored in the black variable
    Call car function
    Reset window display
Exit pygame

```
import pygame
from tkinter import *
from tkinter import messagebox
import sys
import pygame_menu
```

```
window_x = 1000
window_y = 1000
black = (0,0,0)

x =  (window_x/2)
y = (window_y/2)
x_change = 0
y_change = 0
car_speed = 0

race_end = False
rectangle_draging = False
opened = False

car_image = 'images/racecar.png'
racecar = pygame.image.load(car_image)
car_num = 0
car_selection =
['images/racecar.png','images/racecar3.png','images/rsz_racecar2.jpg']

gameDisplay = pygame.display.set_mode((window_x,window_y))

root = Tk()

def close():
    root.destroy()

def pick(selected, value):
    global car_image, car_selection, car_num
    car_num = value
    car_image = car_selection[car_num]

def car(x,y):
    gameDisplay.blit(racecar, (x,y))

def start_the_game():
    global race_end, rectangle_draging,x, y,x_change,y_change ,car_speed,
racecar
    pygame.display.set_caption('Race')
```

```
    racecar = pygame.image.load(car_image)
    racecar.convert()
    rect = racecar.get_rect()
    rect.center = window_x//2, window_y//2
    while not race_end:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                race_end = True
                pygame.quit()
                sys.exit()

            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_LEFT:
                    x_change = -5
                elif event.key == pygame.K_RIGHT:
                    x_change = 5
                elif event.key == pygame.K_UP:
                    y_change = -5
                elif event.key == pygame.K_DOWN:
                    y_change = 5
                elif event.key == pygame.K_TAB:
                    if(not opened):
                        opened = True
                        pick()

            if event.type == pygame.KEYUP:
                if event.key == pygame.K_LEFT or event.key ==
pygame.K_RIGHT:
                    x_change = 0
                if event.key == pygame.K_UP or event.key == pygame.K_DOWN:
                    y_change = 0

            if event.type == pygame.MOUSEBUTTONDOWN:
                if event.button == 1:
                    if rect.collidepoint(event.pos):
                        rectangle_draging = True
                        mouse_x, mouse_y = event.pos
                        offset_x = x - mouse_x
                        offset_y = y - mouse_y
```

```
            if event.type == pygame.MOUSEBUTTONUP:
                if event.button == 1:
                    rectangle_draging = False

            if event.type == pygame.MOUSEMOTION:
                if rectangle_draging:
                    mouse_x, mouse_y = event.pos
                    x = mouse_x + offset_x
                    y = mouse_y + offset_y

        x += x_change
        y += y_change

        if(x >=1000 or y >= 1000):
            # crash
            Tk().wm_withdraw()
            messagebox.showinfo('You crashed BYE BYE')
            pygame.quit()
            sys.exit()

        gameDisplay.fill(black)
        car(x,y)

        pygame.display.update()
pygame.init()

menu = pygame_menu.Menu('Welcome to racing', 400, 300,
                        theme=pygame_menu.themes.THEME_BLUE)

menu.add.text_input('Name:', default='John Doe')
menu.add.selector('Car:', [('1', 0), ('2', 1), ('3',2)], onchange=pick)
menu.add.button('Play', start_the_game)
menu.add.button('Quit', pygame_menu.events.EXIT)
menu.mainloop(gameDisplay)

pygame.quit()
quit()
```

* Car reset allows us to click on the car and reset its position to wherever we drag it to

* We created a new function to update our car image

* We also added press down method to allow us to update our x and y coordinates based on a pressed down mouse click

* At the bottom, we see a new menu item is created. When The Play button is pressed we input the selection of our car and start our game up

Import pygame
Import tkinter to make a graphic user interface
Import sys to exit the program completely
Import pygame_menu to create a game menu

Set x and y sizes to 1000
Set a variable called black to a tuple to represent the color in RGB
Set two variables x and y with half of the values from the window x and y values. (Middle point of the window)
Set x_change and car_speed to 0 for setting the new x position and simulating speed
Set y_change for the new y position
Set race_end to signal the end of the program
Create variables to tell when the rectangle surrounding the car is being dragged to go from keyboard and mouse inputs

Create a variable to check when car selection is opened
Create a variable to know which car has been selected
Create an array of all possible car selections

Initiate pygame
Set game window sizes from x and y variables
Set the title to the window to race
Create a variable to hold an image of a race car

Define a function to close the program
Define the function for picking a car
Define function car
    Update image to a new location
While race hasn't ended
Set car image here
Create a rectangle to surround the car image for dragging
    For every event detected by pygame
        If the event is quit
            End race
        If the event is up, down, right, and left arrows then add or subtract 5 from y or x coordinates

If the tab is pressed open the selection window
If the MOUSEBUTTONDOWN event happens check if the rectangle is touched
If touched update x and y to a new location
Update x and y coordinate
If x or y is greater or equal to 1000 then the car has crashed. Print out the message and end game
Fill background as color stored in the black variable
Call car function
Reset window display
Create a game menu with the title 'Welcome to racing'
Give game menu sections for name, car selection
Give the game menu two buttons Play and Quit. Play button connected to the start of the game.
Exit pygame

Extra:
For extra stuff look here:
https://github.com/DownRamp/Games/blob/master/vroom.py
****
THIS IS THE IMPORTANT PART PLEASE DON'T SKIP
****
Next steps:

- Add an extra car, race against each other
- Make car have speed
- Connect cars (add a function that updates additional image x and y coordinates)