



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS



PROGRAMACIÓN AVANZADA

ASIGNATURA:

Programación Avanzada

PROFESOR:

Ing. Juan Pablo Zaldumbide

PERÍODO ACADÉMICO:

Octubre. 2016 - Marzo. 2017

REPORTE PROYECTO

TÍTULO:

**AVANCE PERSONAL DEL
PROYECTO FINAL**

ESTUDIANTE

Fernanda Ushcasina

FECHA DE REALIZACIÓN: 29 de enero del 2017

FECHA DE ENTREGA: 01 de febrero del 2017

CALIFICACIÓN OBTENIDA:

1 MARCO TEÓRICO

1.1. Introducción al juego “ping pong”

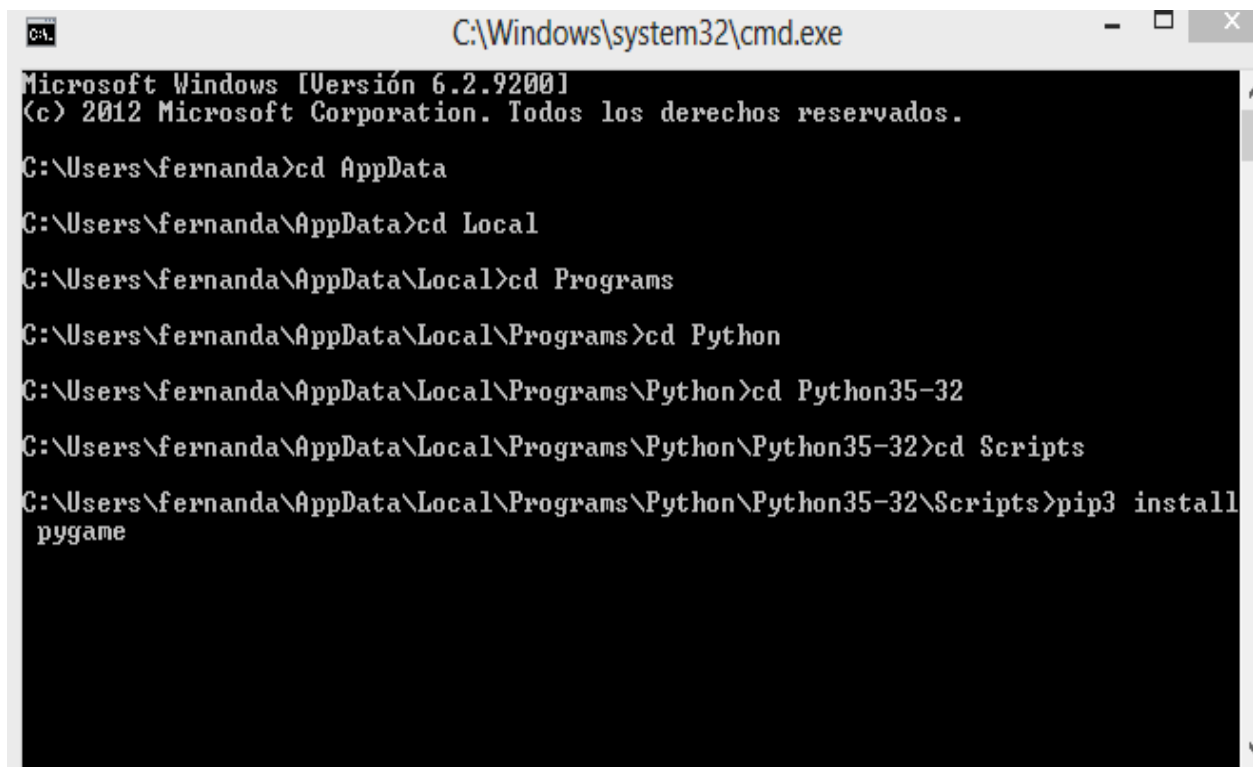
Ping Pong es el famoso videojuego creado por Atari en los 70's, usando el lenguaje de programación Python.

Este es un juego para dos jugadores el cual se trata de que ambos golpeen la pelota con sus respectivas raquetas las cuales deben evitar que la pelota golpee al suelo.

Para programar el juego vamos a usar una librería de Python muy popular llamada Pygame. La cual es muy utilizada ya que es una librería completa y fácil de usar para realizar programas como juegos

Pygame proporciona una API en Python que permite interactuar con la librería multimedia SDL. Pygame es una librería que puede ser utilizada en los principales sistemas operativos (Windows, Mac y Linux).

En esta ocasión hemos instalado Pygame por una línea de comando de la siguiente manera:

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window shows a series of commands being entered to navigate to the Python installation directory and install Pygame. The commands are: "cd AppData", "cd Local", "cd Programs", "cd Python", "cd Python35-32", "cd Scripts", and "pip3 install pygame". The output of the commands is not visible, only the prompts and the commands themselves.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.2.9200]
(c) 2012 Microsoft Corporation. Todos los derechos reservados.

C:\Users\fernanda>cd AppData
C:\Users\fernanda\AppData>cd Local
C:\Users\fernanda\AppData\Local>cd Programs
C:\Users\fernanda\AppData\Local\Programs>cd Python
C:\Users\fernanda\AppData\Local\Programs\Python>cd Python35-32
C:\Users\fernanda\AppData\Local\Programs\Python\Python35-32>cd Scripts
C:\Users\fernanda\AppData\Local\Programs\Python\Python35-32\Scripts>pip3 install
pygame
```

Fig1. (Muestra la instalación de la librería mediante comandos)

2 DESARROLLO

2.1. Definición de las clases las cuales vamos utilizar en el juego.

Vamos a crear las clases que son necesarias para poder realizar el juego.

2.2. Creación de la clase RAQUETAS

Una vez que tengamos lista el área de juego (*creada por Jessica Carrillo*), vamos a programar las raquetas para ambos jugadores. Incluimos el siguiente el siguiente fragmento del código.

```
#programando las raquetas
##clase pad.- hereda los atributos de la clase sprite.Sprite de pygame
class Pad(pygame.sprite.Sprite):
    def __init__(self, pos=(0, 0)):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((12, 30)).convert()
        self.image.fill((255, 255, 255))
        self.rect = self.image.get_rect(center=pos)
        self.max_speed = 10
        self.speed = 0

    def move_up(self):
        self.speed = self.max_speed * -1

    def move_down(self):
        self.speed = self.max_speed * 1

    def stop(self):
        self.speed = 0
##permite actualizar la posicion de la raqueta
    def update(self):
        self.rect.move_ip(0, self.speed)
```

Fig2. (Creación de las raquetas explicación del código)

2.3. Creación de la clase main (llamada a la clase raquetas)

Vamos a crear la clase principal en la cual vamos a la llamar a cada una de las clases, con el fin de que estas empiecen su correcto funcionamiento.

```
def main():

    ##creacion del fondo del juego

    pygame.init()
    size = width, height = 800, 600
    pantalla = pygame.display.set_mode(size)
    pygame.display.set_caption('Pong Pygame')
    salir=False
    player1=Pad() # Instanciamos un objeto de clase Player

    #auxiliares para el movimiento
    izq_apretada, der_apretada, arriba_apretada, abajo_apretada=False, False, False, F
    t=0

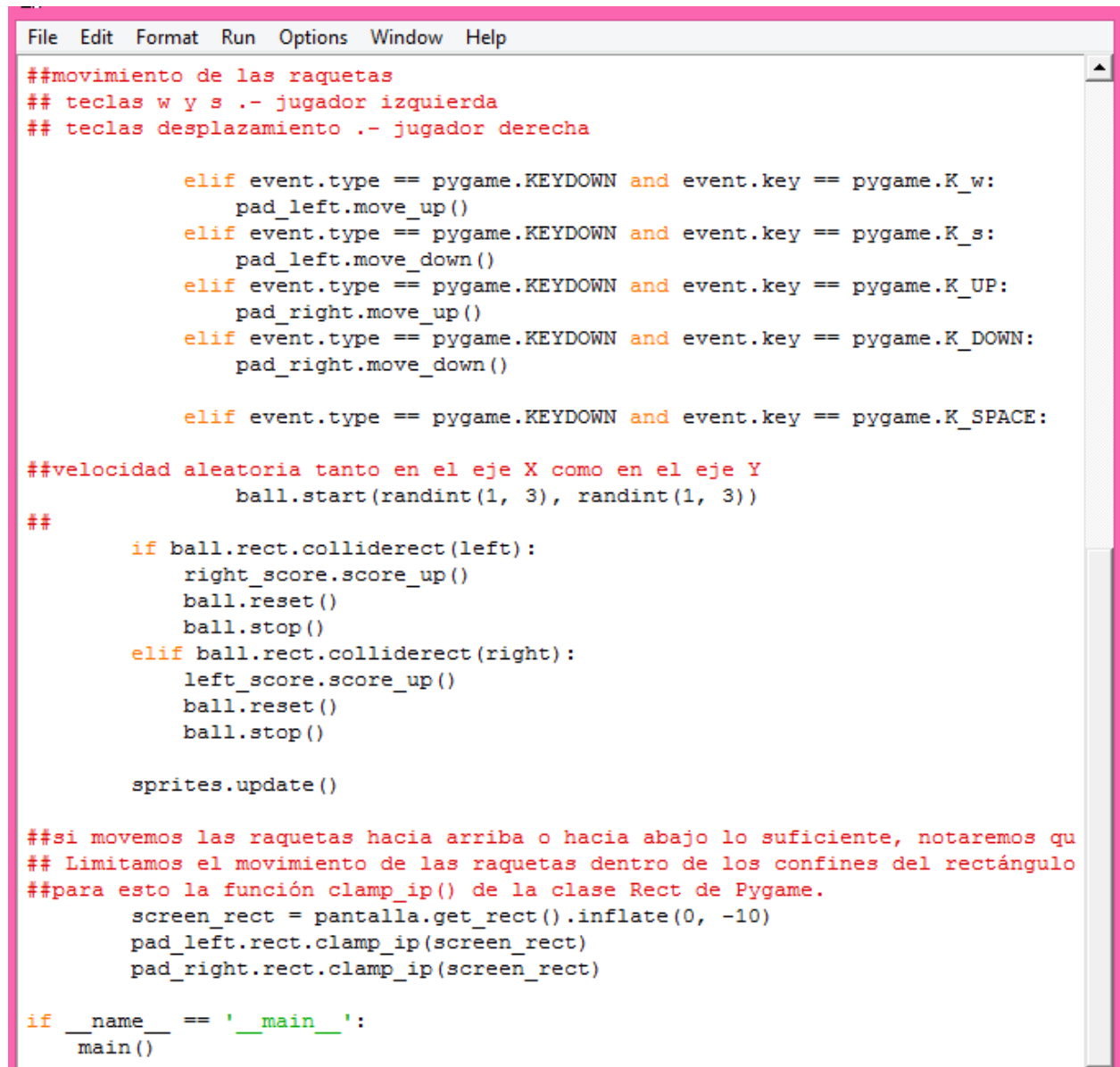
    ##posiciona cada una de las raquetas en la pantalla
    pad_left = Pad((width/6, height/4))
    pad_right = Pad((5*width/6, 3*height/4))
    clock = pygame.time.Clock()
    fps = 60

    ##continúe registrando eventos del teclado mientras mantenemos
    ##presionada una determinada tecla.
    pygame.key.set_repeat(1, fps)

    ##agrupacion sprotos.- raquetas
    sprites = pygame.sprite.Group(pad_left, pad_right)
    sprites = pygame.sprite.Group(pad_left, pad_right, ball, left_score, right_s

    ## bucle infinito
    while 1:

        clock.tick(fps)
    ##hacemos que las raquetas se detengan usando funcion stop
    pad_left.stop()
    pad_right.stop()
```



```

File Edit Format Run Options Window Help

##movimiento de las raquetas
## teclas w y s .- jugador izquierda
## teclas desplazamiento .- jugador derecha

    elif event.type == pygame.KEYDOWN and event.key == pygame.K_w:
        pad_left.move_up()
    elif event.type == pygame.KEYDOWN and event.key == pygame.K_s:
        pad_left.move_down()
    elif event.type == pygame.KEYDOWN and event.key == pygame.K_UP:
        pad_right.move_up()
    elif event.type == pygame.KEYDOWN and event.key == pygame.K_DOWN:
        pad_right.move_down()

    elif event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:

##velocidad aleatoria tanto en el eje X como en el eje Y
        ball.start(randint(1, 3), randint(1, 3))
##
    if ball.rect.colliderect(left):
        right_score.score_up()
        ball.reset()
        ball.stop()
    elif ball.rect.colliderect(right):
        left_score.score_up()
        ball.reset()
        ball.stop()

    sprites.update()

##si movemos las raquetas hacia arriba o hacia abajo lo suficiente, notaremos qu
## Limitamos el movimiento de las raquetas dentro de los confines del rectángulo
##para esto la función clamp_ip() de la clase Rect de Pygame.
        screen_rect = pantalla.get_rect().inflate(0, -10)
        pad_left.rect.clamp_ip(screen_rect)
        pad_right.rect.clamp_ip(screen_rect)

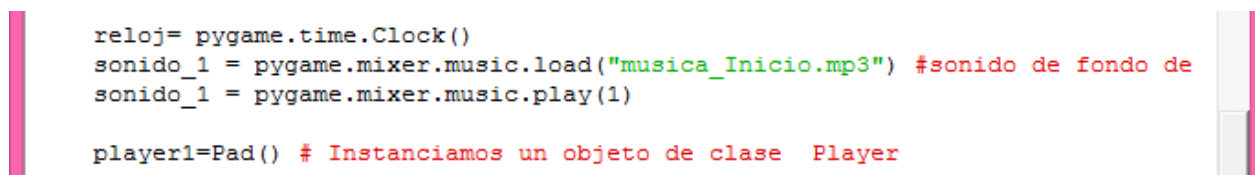
if __name__ == '__main__':
    main()

```

Fig3. (Creación de la clase raquetas con su debida explicación de su funcionamiento.)

2.4. Modificación de la clase main (sonido)

Realizamos esta modificación en la cual para hacer el juego más divertido, colocaremos un código el cual nos permitirá escuchar un sonido, del agrado de los jugadores y su respectivo desempeño a medida que pase le juego.



```

reloj= pygame.time.Clock()
sonido_1 = pygame.mixer.music.load("musica_Inicio.mp3") #sonido de fondo de
sonido_1 = pygame.mixer.music.play(1)

player1=Pad() # Instanciamos un objeto de clase Player

```

Fig4. (Código del sonido)

Cabe mencionar que para que se escuche un sonido tenemos que descargarlo en un archivo mp3, colocamos el nombre de la música y efectivamente la llamamos.

2.5. Ejecución del programa

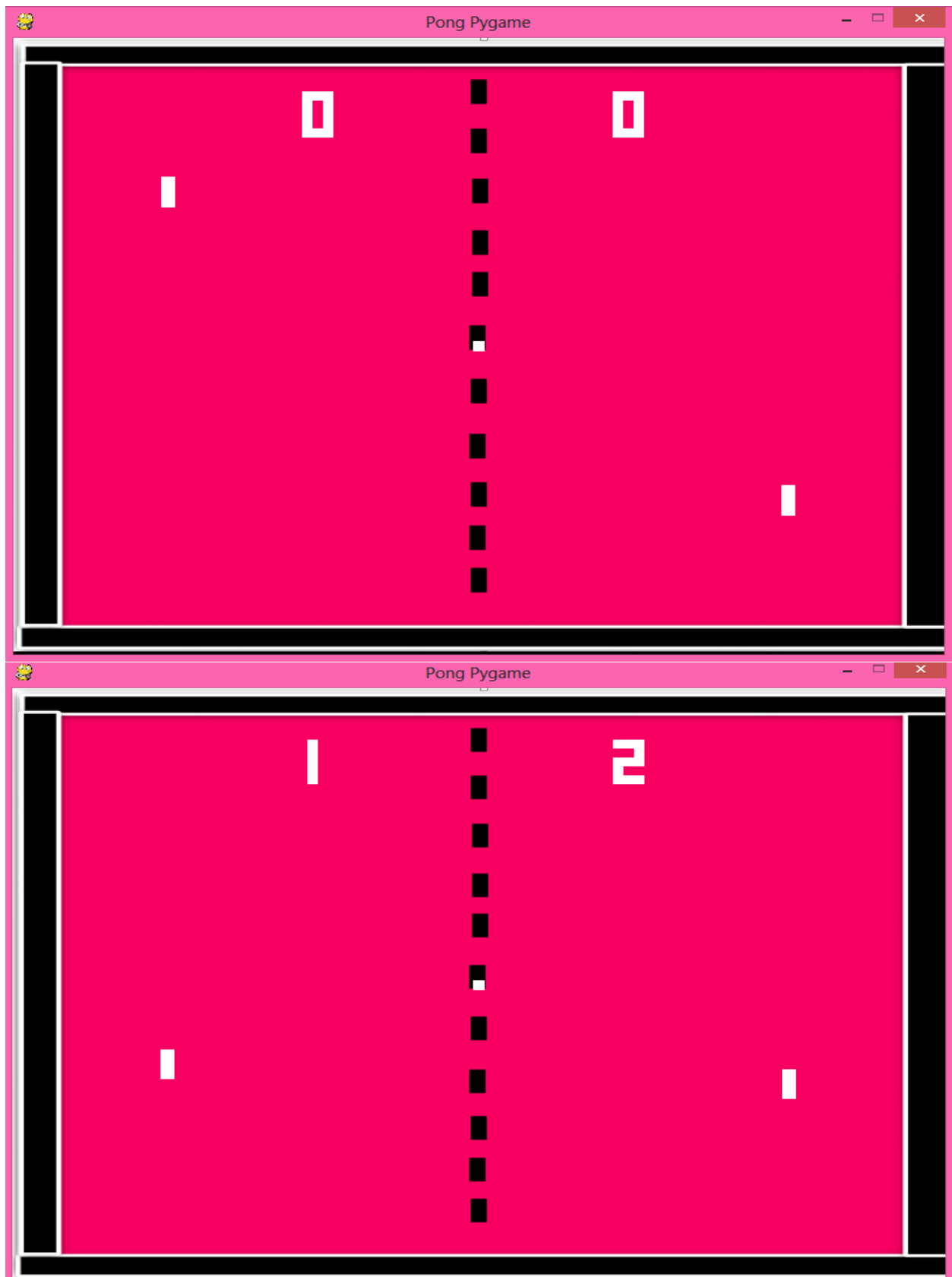


Fig5. (Ejecución del avance del juego, incluye otras clases creadas por diferentes contribuyentes del proyecto.)

3 PROBLEMAS ENCONTRADOS

- Por el momento no hemos encontrado dificultades para realizar el juego, se podría decir que estamos contando con el tiempo y el avance del proyecto tal cual lo hemos programado en nuestro cronograma, el cual mostraremos a continuación.
- Mi compañera de grupo y yo nos ponemos en contacto seguido y tratamos de avanzar en el proyecto lo más rápido posible, a través de investigaciones.

4 CONCLUSIONES

- El avance del proyecto es un éxito y esperamos poder añadirle muchas más cosas al juego que no se ve tan sencillo.
- Estamos poniendo en práctica las enseñanzas que hemos tenido en clase y parecer un poco más a través de las investigaciones.

5 ORGANIZACIÓN DE LAS ACTIVIDADES

En general el trabajo realizado hasta la fecha se distribuyó de la siguiente manera:

Actividad	Responsable	Observaciones
1. Investigación de bibliotecas	todos	
2. Informe: Propuesta del proyecto.	todos	
3. Presentación: Propuesta del Proyecto.	todos	
4. Creación del repositorio.	Jessica Carrillo	
5. Instalación de librerías.	todos	
6. Avance de código: definición de funciones.	todos	
7. Índice del proyecto.	todos	
8. Descripción de los Métodos a implementar.	Fernanda Ushcasina	
9. Función Crear pantalla	Jessica Carrillo	
10. Función: Cargar Imágenes	Jessica Carrillo	
11. Función: clase y movimiento del objeto pelota	Jessica Carrillo	
12. Función: Clase y movimiento del objeto raquetas	Fernanda Ushcasina	
13. Función: Colisiones	todos	
14. Implantación Sonido	Fernanda Ushcasina	
15. Implantación Puntuación	Fernanda Ushcasina	

6 BIBLIOGRAFÍA

[1]"pygame", *Pygame.org*, 2016. [Online]. Available: <http://www.pygame.org/hifi.html>. [Accessed: 30- Jun- 2016].

[2]"[Python] Desarrollo de Interfaces Graficas con Tkinter [Labels,Buttons,Entrys] | Pharalax Blog", *Pharalax.com*, 2012. [Online]. Available: <http://pharalax.com/blog/python-desarrollo-de-interfaces-graficas-con-tkinter-labelsbuttonsentrys/>. [Accessed: 30- Jun- 2016].