**ESCUELA POLITÉCNICA NACIONAL**

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**PROGRAMACIÓN AVANZADA**

|  |  |
| --- | --- |
|  | |
| ASIGNATURA: | Programación Avanzada |
| PROFESOR: | Ing. Juan Pablo Zaldumbide |
| PERÍODO ACADÉMICO: | Oct. 2016- Marz. 2017 |
|  | |

**INFORME PROYECTO FINAL**

|  |  |
| --- | --- |
|  | |
| TÍTULO:  **PING-PONG** | |
| **ESTUDIANTE** | |
| Jessica Carrillo, Fernanda Ushcasina | |
|  | |
|  | |
|  | |
|  | |
|  | |
| FECHA DE REALIZACIÓN: | 12 de Febrero del 2017 |
| FECHA DE ENTREGA: | 15 de Febrero del 2017 |
|  | |

Contenido

[1. INTRODUCCION 2](#_Toc474919426)

[2. OBJETIVOS GENERALES 3](#_Toc474919427)

[3. OBJETIVOS ESPECIFICOS 3](#_Toc474919428)

[4. MARCO TEORICO 3](#_Toc474919429)

[5. PROGRAMAS UTILIZADOS 5](#_Toc474919430)

[6. MANUAL DE INSTALACION 5](#_Toc474919431)

[Python: 5](#_Toc474919432)

[Pygame 8](#_Toc474919433)

[7. ACTIVIDADES REALIZADA POR LOS INTEGRANTES 9](#_Toc474919434)

[Jesica Carrillo y Fernanda Ushcasina 9](#_Toc474919435)

[8. CRONOGRAMA DE DESARROLLO DE ACTIVIDADES 16](#_Toc474919436)

[9. PROBLEMAS ENCONTRADOS 17](#_Toc474919437)

[10. CONCLUSIONES 17](#_Toc474919438)

[11. RECOMENDACIONES 17](#_Toc474919439)

[12. Bibliografía 18](#_Toc474919440)

# INTRODUCCION

Ping-Pong juego de mesa que es ejecutado por dos jugadores el mismo que consta de dos raquetas y una pelota con el objetivo de que cruzar la barrera que el oponte posee y al momento de tocar la pared o barrera se adquiera un punto y el mismo termina con una puntuación sobre 11. Para la elaboración de los elementos, movimientos y estrategias implementadas a este proyecto son la elaboración de distintas clases las mismas que poseen funciones que permitirán que los elementos interactúen con el usuario dentro de una cuadro con dichas dimensiones además estas clases pueden llamadas las veces que se necesite facilitando la programación ya que se manejan a través de herencia por los atributos dados.

# OBJETIVOS GENERALES

* Crear un juego, poniendo en práctica todo lo aprendido durante el periodo de clases y cumplir con las especificaciones que se piden.
* Implementación de un menú a través del cual podamos elegir las opciones de juego

# OBJETIVOS ESPECIFICOS

* Mostrar el perfecto funcionamiento del juego
* Crear un juego en el cual podremos interactuar, tanto con la maquina como con nosotros mismo (2 jugadores)
* Crear programa, el mismo que sea fácil de entender para los usuarios.
* Mostrar la carga de imágenes, colisiones, música, puntajes.
* Realizar movimientos de las raquetas, pelota.
* Mostrar el correcto uso de las librerías necesarias para realizar el juego.
* Creación de clases como paletas y pelota, botón, cursor.
* Declaración de funciones como: cargar imágenes, actualizar principal.
* Manejo de imágenes formato PNG, como también audio de formato mp3.

# MARCO TEORICO

En este proyecto final hemos implementado todos los conocimientos adquiridos en clase, con el fin de poder desarrollar un juego interactivo, interfaz amigable y de fácil uso para los usuarios.

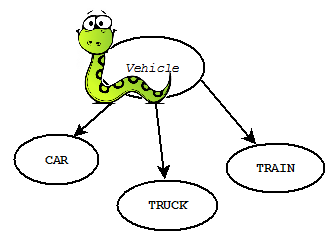
Para poder realizar un juego con la ayudad de la librería Pygame, por lo que veremos algunos términos necesarios para entender el trabajo que se realizó.

* **Pygame** es un motor de juegos, conformado por un conjunto de librerías cuya finalidad es facilitar la tarea del programador a la hora de realizar un videojuego, pudiendo ser usado también para la realización de aplicaciones multimedia e interfaces gráficas.



*Fig1. (Imagen juego Pygame)*

* **Clases**: las clases en Python son un mecanismo, fácil para programar. Es otras palabras una clase es la clasificación de un objeto. Un objeto es una instancia de la clase en particular creada, y estos objetos posen atributos que pueden ser el nombre, tamaño, imagen y de la misma manera se puede dar diferentes métodos que se definen como las acciones que realizaran los objetos.



*Fig2. (Instanciando una clase)*

* **Librería random**: Modulo random en Python con ejemplos. Funciones que ellos contienen, vistos en el correr del blog. El módulo random proporciona un generador de números aleatorios, por lo que es adecuado para una gran gama de aplicaciones: juegos, web, bases de datos, etc.
* **Modulo os**: El módulo os nos permite acceder a funcionalidades dependientes del Sistema Operativo. Sobre todo, aquellas que nos refieren información sobre el entorno del mismo y nos permiten manipular la estructura de directorios (para leer y escribir archivos)
* **Módulo sys**: El módulo sys es el encargado de proveer variables y funcionalidades, directamente relacionadas con el intérprete.

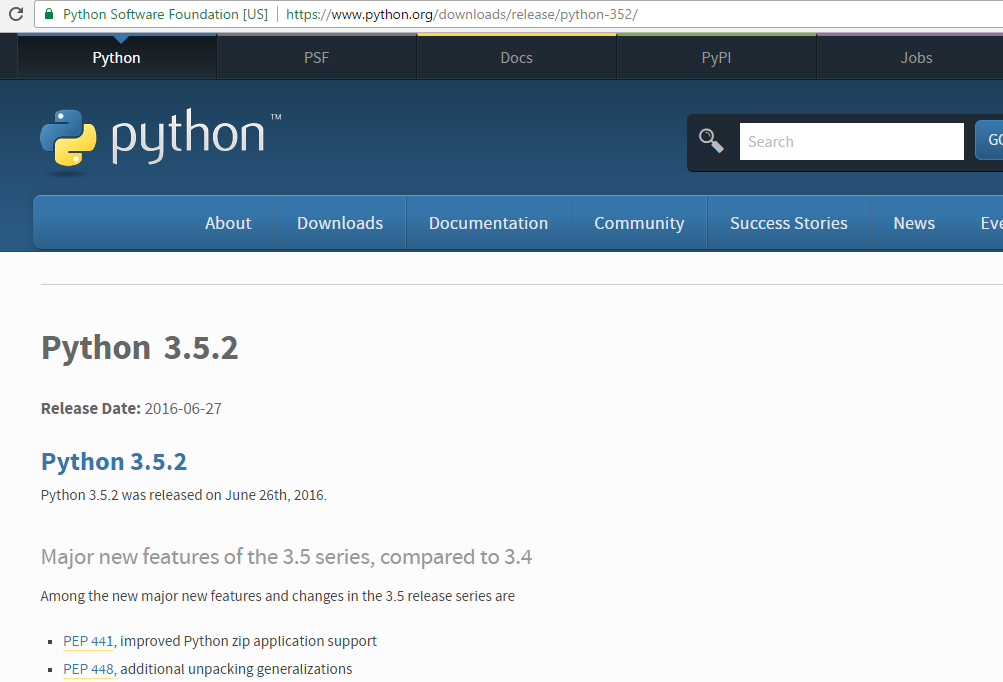
# PROGRAMAS UTILIZADOS

* Python
* Librerías:
* Pygame
* os
* Sys (instalada por defecto)
* Random (instalada por defecto)

# MANUAL DE INSTALACION

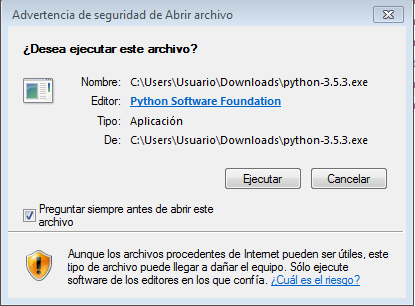
## Python:

* Para su instalación nos dirigimos a su página principal, y descargamos la versión que queramos, dependiendo de nuestro sistema operativo.



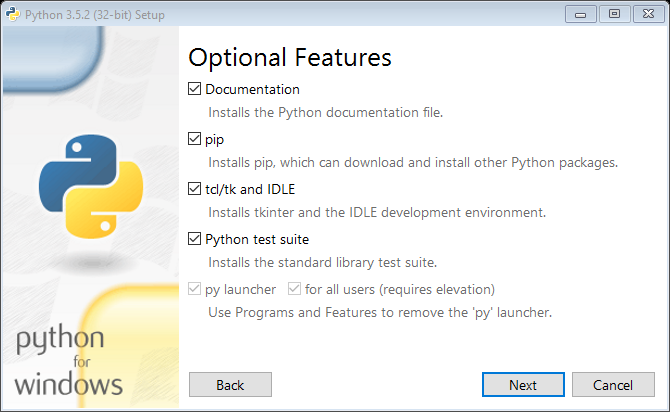
*Fig3. (Pagina principal)*

* Descarga finalizada del programa (ejecutable), clic en ejecutar.



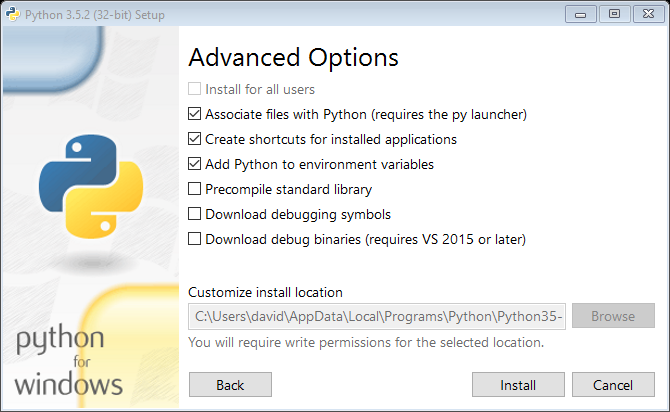
*Fig4. (Ejecutable del programa)*

* Una vez ejecutado nos muestra la siguiente pantalla, clic en next.



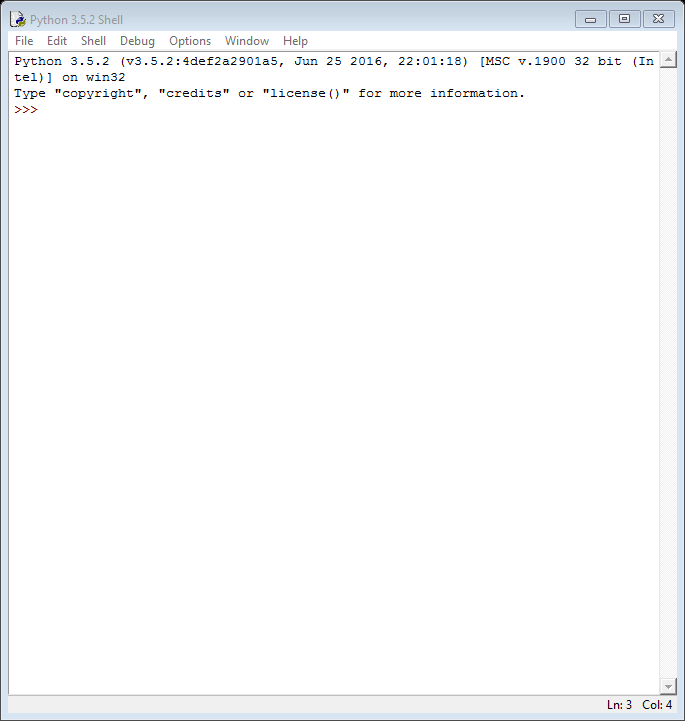
*Fig5. (Instalación de Python)*

* Seleccionamos las casillas ADDPATH y clic en install.



*Fig6. (Instalando Python)*

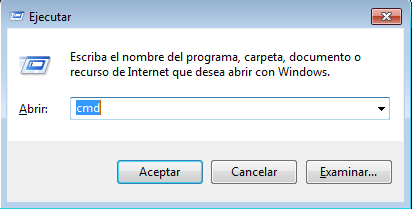
* Una vez finalizada la instalación, nos muestra este editor.



*Fig7. (Pantalla principal de editor)*

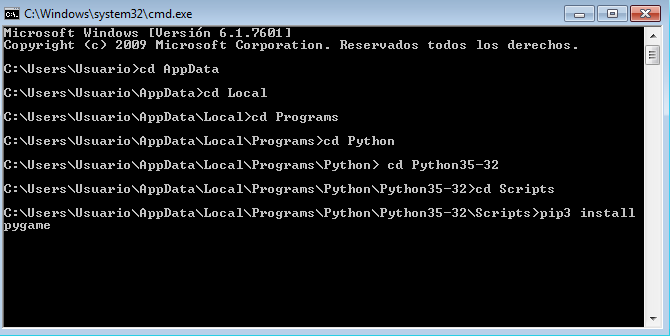
## Pygame

* Para la instalación de la librería Pygame, en nuestro caso lo realizaremos mediante líneas de comando, para lo cual necesitamos abrir el Símbolo del sistema (cmd digitando las teclas \*Windows + r\*)



*Fig8. (Símbolo del sistema)*

* Una vez en el *cmd*  empezamos abrir la ruta de Python para poder instalar el programa Pygame.



*Fig9. (ingresando a la ruta de Python para instalar Pygame)*

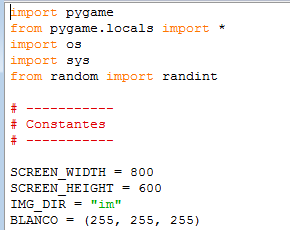
* Una vez finalizada la instalación ya se podrá importar la librería de pygame.



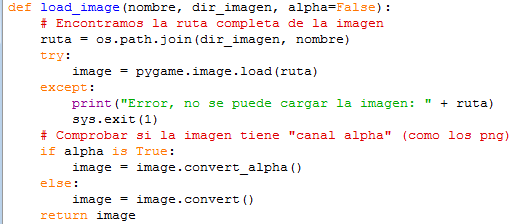
# ACTIVIDADES REALIZADA POR LOS INTEGRANTES

## Jesica Carrillo y Fernanda Ushcasina

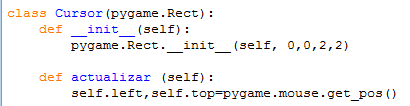
* Empezamos por importar las librerías necesarias para la ejecución del juego.
* Contantes:
  + Ancho de la pantalla
  + Alto de la pantalla
  + Ruta donde se encuentran las imágenes guardadas.
  + Colores de la pantalla.



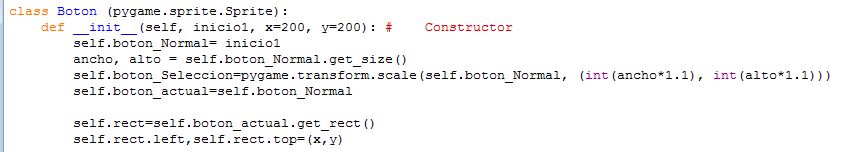
* Cargas de imágenes:
  + Definición de las clases para poder cargar las imágenes, misma que comprobara su ruta para poder mostrarlas.



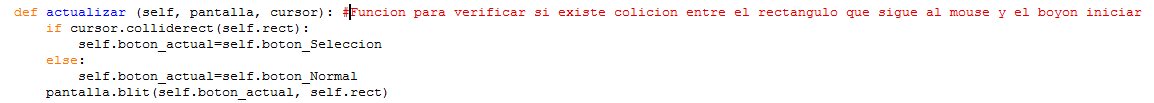
* Clase cursor:
  + Nos permitirá el movimiento del curso para poder seleccionar el tipo de juego que queramos.
  + Jugar contra CPU jugar entre dos jugadores.



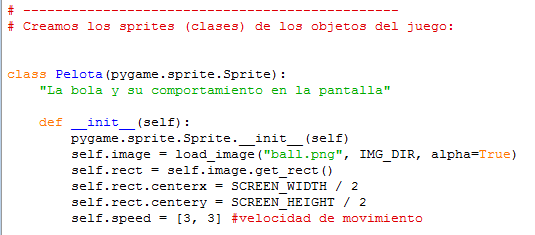
* Clase Botón:
  + Nos permite el inicio del juego.



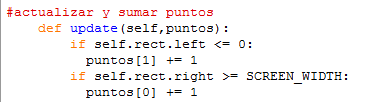
* Actualizar:
  + Actualiza el juego y nos muestra
  + Verifica las colisiones existentes.



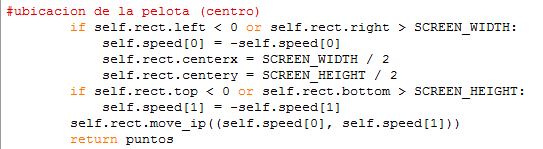
* Clase pelota:
  + Esta clase nos muestra el comportamiento de la pelota en el juego.
  + Los movientes de la pelota gracias a los sprites.
  + Nos muestra la velocidad de la pelota la cual podemos editarla.
  + Colisiones de la pelota dentro de la pantalla de juego.



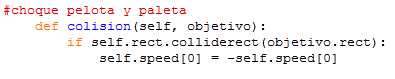
* Actualizar y sumar puntos:
  + Esta función nos permite sumar los puntos que realicemos al momento de la jugada.



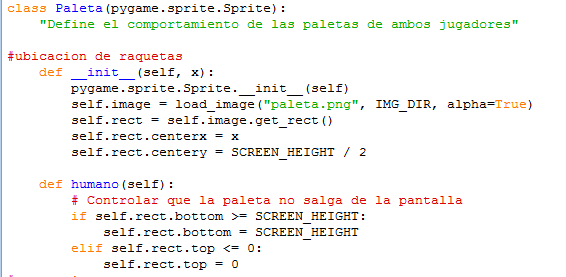
* Ubicación de la pelota:
  + En estas líneas de código podemos ver la ubicación de la pelota, en este caso la misma que rebotara desde el centro de la pantalla, una vez que se presione la tecla ESPACIADORA empezar el movimiento.



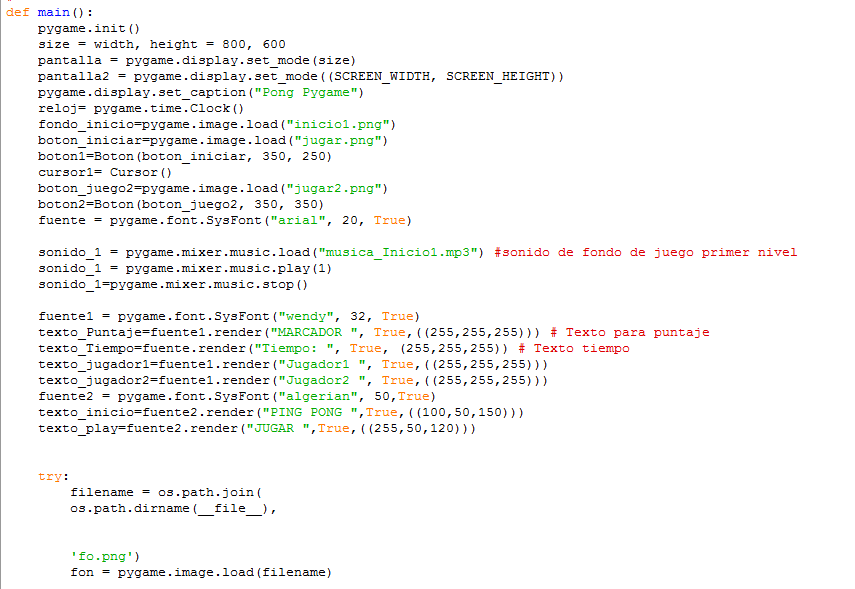
* Colisiones:
  + Colisiones de cada jugador.

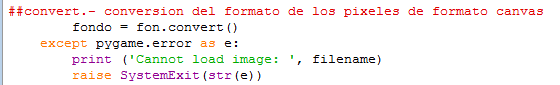


* Clase Paleta:
  + Aquí mostramos la ubicación de las paletas, tanto de la maquina como de los jugadores.

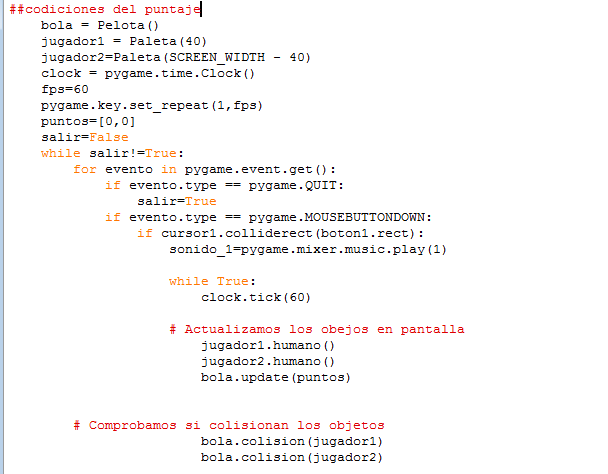


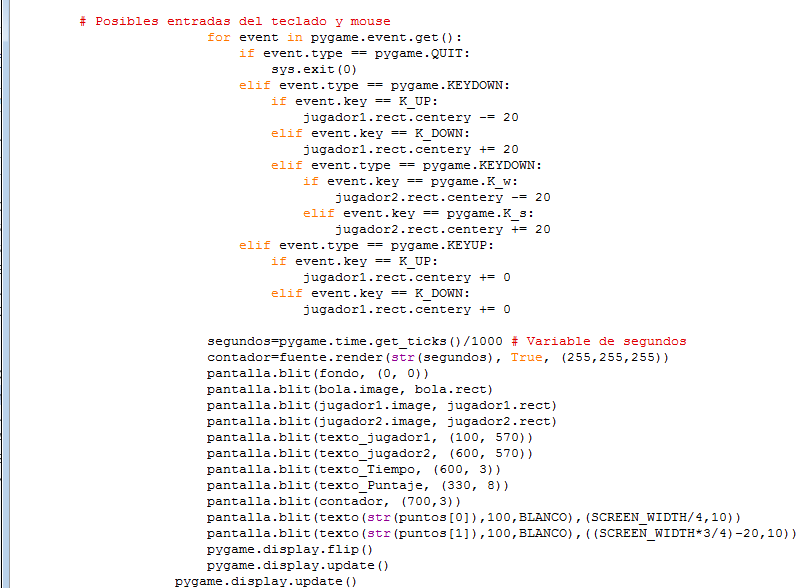
* Función main:
  + Esta es la clase principal en la cual podemos llamar a varias funciones para el funcionamiento del programa.
  + Carga de imágenes.
  + Carga de sonidos.
  + Llamado de fuentes para los respectivos textos.
  + Colores para las pantallas, textos.

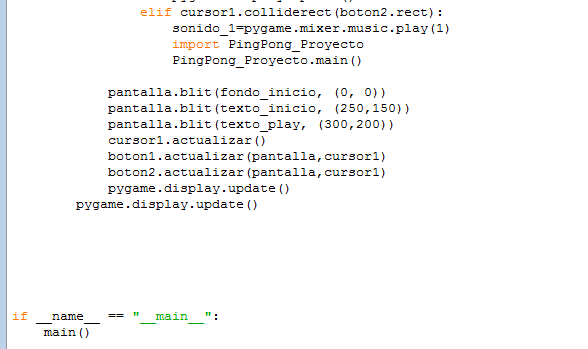




* Condiciones del puntaje:
  + El puntaje aumenta a través de los golpes que de la raqueta a la pelota, si esta no puede golpearle será punto para el otro jugador y así sucesivamente.







* Ejecutando el programa.





# CRONOGRAMA DE DESARROLLO DE ACTIVIDADES

Todos los integrantes del grupo hemos cumplido las indicaciones según el cronograma que se ha realizado con anterioridad y finalmente se ha finalizado con lo programado.

|  |  |  |
| --- | --- | --- |
| Actividad | Responsable | Observaciones |
| 1. Investigación de bibliotecas | todos |  |
| 1. Informe: Propuesta del proyecto. | todos |  |
| 1. Presentación: Propuesta del Proyecto. | todos |  |
| 1. Creación del repositorio. | Jessica Carrillo |  |
| 1. Instalación de librerías. | todos |  |
| 1. Avance de código: definición de funciones. | todos |  |
| 1. Índice del proyecto. | todos |  |
| 1. Descripción de los Métodos a implementar. | Fernanda Ushcasina |  |
| 1. Función Crear pantalla | Jessica Carrillo |  |
| 1. Función: Cargar Imágenes | Jessica Carrillo |  |
| 1. Función: clase y movimiento del objeto pelota | Jessica Carrillo |  |
| 1. Función: Clase y movimiento del objeto raquetas | Fernanda Ushcasina |  |
| 1. Función: Colisiones | todos |  |
| 1. Implantación Sonido | Fernanda Ushcasina |  |
| 1. Implantación Puntuación | Fernanda Ushcasina |  |

# PROBLEMAS ENCONTRADOS

* Uno de los problemas que tuvimos fue la compatibilidad de python.

# CONCLUSIONES

* La programación de video juegos utilizando el programa Python es muy fácil, ya que Python al hacer multiplataforma en este se puede instalar motores librerías que facilitan en mucho la programación.
* Con todo esto los programadores pueden realizar diferentes tipos de video juegos y orientar su producto a usuarios en específicos o en una porción del mercado de los videos juego, brindando un producto de calidad con estándares de calidad, con la existencia de este motor llamado Pygame toda persona con un poco de conocimiento en programación puede realizar un juego, diversificando el mercado de los videos juego y que el usuario tenga la protesta de elegir diferentes productos.
* Los conocimientos adquiridos en clase que fueron impartidos por el profesor, fueron útiles para desarrollar nuestro proyecto ya que toda la infraestructura utilizada y el funcionamiento del mismo es similar utilizando el motor Pygame, además dentro de este podemos combinar Pygame y las otras librerías utilizadas en clase como Turtle, Sys etc.
* La utilización de Pygame como el principal instrumento que tomamos para realizar todo nuestro proyecto ya que es el motor más fácil de utilizar y es exclusivamente orientado a la creación de video juego ya que cuenta con diferentes librerías para el mismo, incluye funciones para facilitar la utilización de archivos de multimedia en todo parte del código.

# RECOMENDACIONES

* Para poder utilizar correctamente Pygame debemos tener conceptos básicos sobre programación en Python esto nos ayudara a una implementación más rápida y entendible.
* Realizar clases con los parámetros necesarios para que funcione correctamente en la carga de imágenes, datos y música y Utilizar imágenes con formato png para tener mayor facilidad al momento de tener transparente el fondo de la imagen.
* Antes de empezar a realizar el proyecto, tenemos que tener bien en cuenta la lógica del juego y los recursos necesarios que vamos a utilizar para realizarlo de manera correcta.
* Manejar excepciones al momento de trabajar con archivos texto en donde si no se encuentren dichos archivos el programa los cree automáticamente y que el programa trabaje sobre ellos evitando errores, pero tomando en cuenta que si eliminan estos archivos se perdería lo que es la información de los contenidos logrados.

# Bibliografía

[1]"display [Losersjuegos]", *Losersjuegos.com.ar*, 2016. [Online]. Available: http://www.losersjuegos.com.ar/traducciones/pygame/display. [Accessed: 21- Jul- 2016].

[2]"Editor Audio Mp3 Online", *Editor Audio Mp3 Online*, 2016. [Online]. Available: http://mp3.editaraudio.com/. [Accessed: 10- Aug- 2016].

[3]"Tutorial Pygame", *Sabia.tic.udc.es*, 2016. [Online]. Available: http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/ProgramacionVideoJuegos/PyGame/python.html. [Accessed: 10- Aug- 2016].

[4]P. Craven, "Programar Juegos Arcade con Python y Pygame", *Programarcadegames.com*, 2016. [Online]. Available: http://programarcadegames.com/index.php?chapter=introduction\_to\_sprites&lang=es. [Accessed: 10- Aug- 2016].

[5]"Pygame VII: Colisiones | Razón Artificial", *Razonartificial.com*, 2016. [Online]. Available: http://razonartificial.com/2010/02/pygame-7-colisiones/. [Accessed: 10- Aug- 2016].

[6]"Agregar Sonido desde la Biblioteca", *Aprendiendo a Programar en Python con PyGame para hacer Video Juegos*, 2006. [Online]. Available: https://miprogramacionenjuegos.wordpress.com/2006/12/08/agregar-sonido-desde-la-biblioteca/. [Accessed: 10- Aug- 2016].

[7]"Reloj Digital - Wiki PyAr - Python Argentina", *Python.org.ar*, 2016. [Online]. Available: http://www.python.org.ar/wiki/RelojDigital. [Accessed: 10- Aug- 2016].