# Videojuego Controlado por Puerto Serial

El objetivo del proyecto fue programar un videojuego controllable por información recibida via puerto serial. Para ello se utilizó el lenguaje de programación Python y las librerías “binascii”, “os”, “sys”, “serial” y “Pygame”. A continuación se detalla la filosofía de diseño utilizada y el esquema general del código.

## Esquema de Diseño

El esquema de diseño divide el código en tres partes principales:

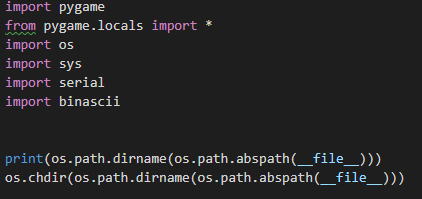
1. Configuración inicial de parámetros.
2. Adquisición de data.
3. Ejecución de la aplicación.

### Configuración General de Parámetros

En esta etapa se importaron las librerías a utilizar y se configuraron las variables generales a usar en el juego. Esto incluye la creación e inicialiazción de los objetos “ship” y “stone” que representan al jugador y los obstáculos del juego respectivamente.

Adicionalmente, se incluyen las funciones para los objetos “player” “obstacle” “projectile” y “redrawWindow” basadas en los tutoriales para programación disponibles gratuitamente en el canal de YouTube “Tech With Tim”[1].

En la siguiente figura se puede apreciar el código para importar las imágenes y actualizar el directorio al local donde se encuentre guardado el código.



Es importante actualizar el directorio para asegurarse que el programa pueda encontrar las imágenes necesarias para la ejecución de la aplicación sin errores.

Finalmente, En la inicialización del programa se configuran parámetros generales para la ejecución de la aplicación como son:

* El tamaño de la ventana de la aplicación.
* Coordenadas base para el jugador.
* Variables de contéo y auxiliares.
* Carga de imágenes para el jugador y fondo de juego que se pueden apreciar en las siguientes figuras.





### Adquisición de Data

La adquisición de data serial para controlar el jeugo se realizó mediante la aplicación de dos librerías:

* “serial” para configurar el objeto puerto, abrirlo, adquirir los datos entrantes y flushearlo periódicamente.
* “binascii” para poder procesar los bytes de data y realizar el desenamscaramiento de la información asíncrona.

La trama entrante llega en 5 bytes de la siguiente forma:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | Data 1 | 1 | Data 2 | 1 | Data 3 | 1 | Data 4 | 1 | d1,2,3 | a | b | c | d |
| Byte 1 | | Byte 2 | | Byte 3 | | Byte 4 | | Byte 5 | | | | | |

Donde el 0 en el bit más significativo del byte 1 marca la diferencia apra reconocimiento con respecto a los 1 de los demás bytes. "d1,2,3” son 3 bits reservados para 3 posibles canales digitales de los que se utilizaron 2.

La data luego de ser desentramada debe quedar de la siguiente manera:

|  |  |  |  |
| --- | --- | --- | --- |
| A | Data 1 | b | Data 2 |
| Canal Analógico 1 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| c | Data 3 | d | Data 4 |
| Canal Analógico 2 | | | |

El desentramado y adquisición se realizó mediante la función “serialGet(puerto,flush timer)” que recibe como entrada el objeto puerto por el que llega la data y un valor de contador predeterminado que indica si debe realizar un flusheo al puerto serial.

Como salidas la función retorna:

* Un entero que depende de la data del canal analógico 1.
* Un entero que depende de la data del canal analógico 2.
* El valor del canal digital 1.
* El valor del canal digitall 2.
* El contador de flusheo recibido que puede estar reseteado a su valor inicial o con 1 menos dependiendo de si se realizó el flusheo o no.

### Ejecución de la Aplicación

La aplicación se ejecuta en un ciclo while que se mantiene mientras la variable “run” se mantenga como “True” en ella se:

1. Crean y mueven los proyectiles, obstáculos, fondo y personaje del jugador.
2. Verifica si coinciden las coordenadas de los obstáculos con el jugador o proyectiles para terminar el juego o eliminar el obstáculo/proyectil y aumentar el score respectivamente.
3. Adquiere data de serial llamando la función serialGet().
4. Verifica según la data adquirida si el jugador “salta”, “dispara”, “pone en pausa el juego” o lo “cierra” mediante los valores recibidos por el canal analógico 1, canal analógico 2, canal digital 1 y canal digital 2 respectivamente.

[1] Tech With Tim. (2019). Retrieved from

https://www.youtube.com/channel/UC4JX40jDee\_tINbkjycV4Sg/about