

## Nivel Secundario

### Programación en Python - Desafío de la Ronda 2

Felicitaciones equipo, muy buen trabajo, han cumplido con el desafío inicial. Ahora pasaremos de nivel para emprender aventuras aún más desafiantes.

¡COMENCEMOS!

#### Introducción

A medida que la población aumenta se requiere una mejor administración de los paquetes a entregar. Por este motivo, se considera importante mejorar las formas de colaboración con los robots para satisfacer las necesidades de los seres humanos.

#### Objetivo general de la actividad

Ahora que ya programamos los elementos básicos para manejar a UAIBOT, les comentamos que se suma otro desafío: trabajo en equipo.

UAIBOT, UAIBOTA, UAIBOTINA y UAIBOTINO deberán colaborar con la entrega de paquetes.

¿Están listos para otro desafío? Manos a la obra...

#### Desafío

Mejorar el desarrollo del juego para permitirle al jugador, representado por UAIBOT y su familia, conseguir baterías para realizar sus tareas eficientemente.

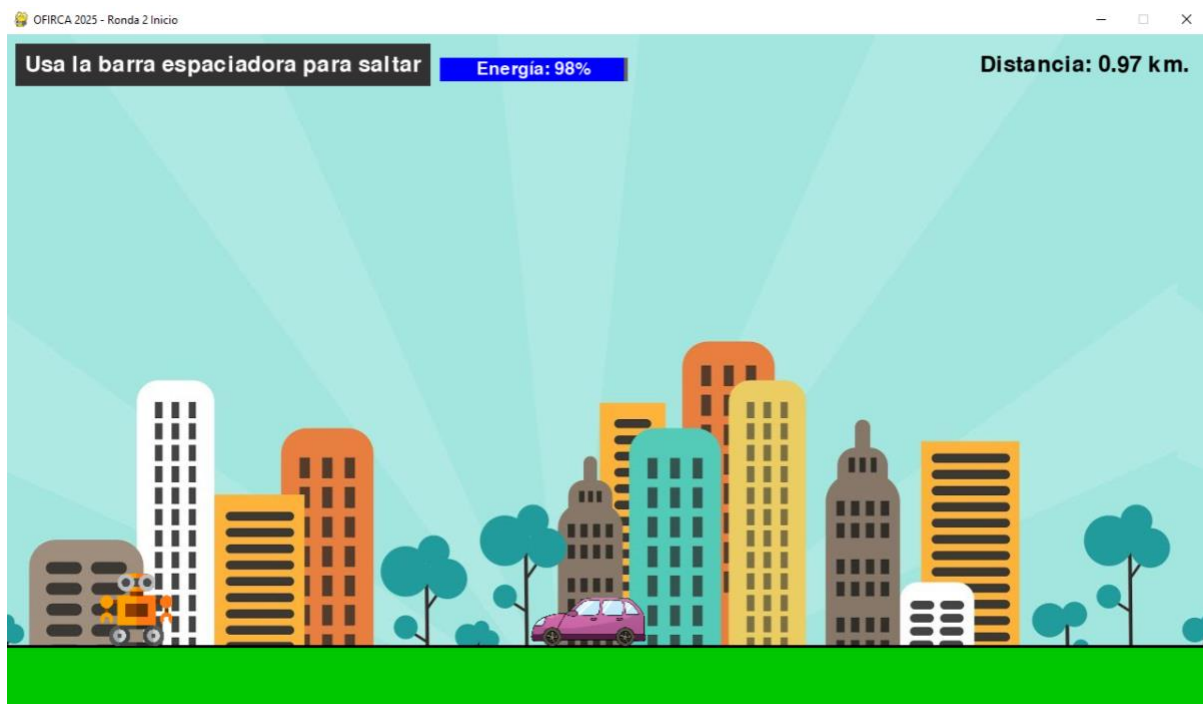
Es fundamental para ello, tener en cuenta las características del juego, los requerimientos del desarrollo y las consignas específicas de trabajo que se enuncian a continuación.

Los proyectos ganadores de esta olimpiada serán compartidos con estudiantes de escuelas de nivel primario. Buscamos que cada juego pueda ser recorrido e intervenido por quienes quieran contribuir a enriquecer la propuesta, y crear nuevas ideas a partir de lo que programaron.

## Características del juego

- a. El escenario expuesto consiste en un área urbana con UAIBOT y un auto.
- b. La interfaz gráfica puede mostrar cuándo termina el juego si UAIBOT colisiona contra un auto o si el paquete fue entregado con éxito al recorrer un kilómetro y medio. Además, existe un contador de kilómetros recorridos.

A continuación, se ilustra la interfaz gráfica:



## Requerimientos sobre los cuales se basó el juego que les facilitamos

Con objeto de facilitarles la lectura y la comprensión del código del juego, les detallamos los requerimientos sobre los cuales se basó su arquitectura.

1. En la etapa inicial se realizó:
  - 1.1. El desarrollo del código que permite:
    - 1.1.1. Reemplazar la imagen del cuadrado y rectángulo por el de UAIBOT y un auto.
    - 1.1.2. Hacer que UAIBOT salte con la barra espaciadora.
    - 1.1.3. Mostrar la barra de energía que dura 60 segundos con el porcentaje dentro, el contador de kilómetros decrementándose y el texto de "juego terminado" con la pausa del juego correspondiente al haber un choque entre UAIBOT y el auto.

2. En la etapa avanzada se realizó:

2.1. El desarrollo del código que permite:

2.1.1. Mostrar con un texto que el paquete se entregó exitosamente luego de recorrer un kilómetro.

2.1.2. Hacer que el auto vuelva a aparecer cuando se va del escenario.

2.1.3. Animar el fondo para lograr una sensación de velocidad.

Incorporar los ajustes sugeridos en “las consignas específicas de trabajo” es muy importante, pero fundamentalmente se valorarán aquellos aportes creativos que permitan resolver exitosamente el desafío con originalidad.

## Consignas específicas de trabajo

A partir del juego que le facilitamos, incorporen los siguientes ajustes al juego:

1. Habilitar el cambio de personaje con la letra C únicamente cuando el robot está sobre el piso. El jugador deberá poder elegir entre UAIBOTA, UAIBOTINA y UAIBOTINO. Se pueden usar las imágenes provistas u otras.
2. Diferenciar las alturas de los saltos para cada uno de los robots. UAIBOTA deberá saltar más alto que todos. UAIBOT deberá saltar más alto que UAIBOTINA y UAIBOTINO.
3. Hacer que cada robot tenga una cantidad máxima diferente de segundos de autonomía. UAIBOT=20 segundos. UAIBOTA= 10 segundos. UAIBOTINA= 30 segundos. UAIBOTINO= 30 segundos. Dependiendo del robot elegido, la barra de energía deberá mostrar el porcentaje de energía restante actual para el robot seleccionado.
4. Agregar pilas en el escenario a una altura aleatoria para que, al recolectar cada una, otorgue 10 segundos de autonomía adicionales para el robot elegido. Al recolectar una pila, únicamente se cargará la barra de energía del robot que la recolecte sin superar su autonomía máxima.
5. El juego también termina cuando alguno de los robots se queda sin energía.
6. El juego debe poder reiniciarse con la tecla R una vez que termina. Indicarlo con una instrucción sobre fondo gris oscuro que parpadee en forma intermitente y que su texto cambie de color al azar cada vez que aparece.
7. Reproducir un sonido cuando alguno de los robots salte y cuando se reinicie el juego.
8. Mostrar en la interfaz gráfica la distancia que está recorriendo cada robot. El contador de km. de cada robot aumenta cuando el robot es seleccionado. Al reiniciar el juego, se guarda en un archivo cuántos kilómetros recorrió cada robot en cada partida. Las partidas deben numerarse en forma consecutiva. Guardarlo de la siguiente manera utilizando comas en cada número:

## Partida 1:

UAIBOT: 0,18 km

UAIBOTA: 0,13 km

UAIBOTINA: 0,15 km

UAIBOTINO: 0,06 km

## Partida 2:

UAIBOT: 0,14 km

UAIBOTA: 0,25 km

UAIBOTINA: 0,00 km

UAIBOTINO: 0,00 km

## Partida 3:

UAIBOT: 0,27 km

UAIBOTA: 0,03 km

UAIBOTINA: 0,05 km

UAIBOTINO: 0,19 km

9. Se fomenta el uso de Inteligencia Artificial para realizar los desafíos. En caso de usarla, agregar un archivo de texto .txt a la solución con los prompts utilizados.
10. Cualquier funcionalidad extra sumará muchos puntos en la evaluación del jurado siempre y cuando estén cumplidas las consignas anteriores. El jurado presta mucha atención a la creatividad. Indicar en un archivo de texto .txt qué se agregó.
11. Cuando finalicen las consignas del desafío guarden su proyecto de manera local y luego, suban el proyecto en la "Plataforma de OFIRCA" en formato ZIP, comprimiendo todos los archivos necesarios para su corrección. ¡Éxitos!

## Conceptos y herramientas que pueden utilizar

- IDE para Python 3.12.3 IDLE (entorno de desarrollo local)
- Uso de librerías.
- Ingreso de datos por teclado
- Uso de estructuras secuenciales, condicionales (if..else) y de repetición (while, for).
- Estructuras de datos
- Definición y uso de funciones
- Manejo de archivos
- Uso de imágenes y teclado para animar objetos





## Recursos multimedia

Para la realización de esta actividad se podrá utilizar material multimedia generado por el equipo participante, o aquel que cuente con derechos de autor con licencia Creative Commons que permita su reutilización de forma gratuita.

## Criterios de evaluación

- **Respeto por las consignas** sobre las cuales se basó el juego que les facilitamos.
- **Legibilidad** en los **códigos** y **claridad** en los **comentarios** representativos de la funcionalidad o tarea que están describiendo.
- **Claridad en la definición de los nombres** y significados de las **variables** y **funciones**.
- **Coherencia estética** y **facilidad de uso** del juego.
- **Creatividad y originalidad** en las propuestas de soluciones.
- **Uso responsable de la Inteligencia Artificial. Documentación del uso de la Ingeniería de Prompts.**

