



# Proyecto Final

## Videojuego en Python

### "AtropellaPollos"



**Departamento de Ciencias  
de la Computación**

**Asignatura:**

*"Lenguajes Inteligentes"*

**Profesor:**

Alejandro Padilla Díaz

**Fecha:**

*5 de diciembre de 2024*

**Alumnos:**

- Juan Francisco Gallo Ramírez
- Pablo Emilio Soto Parada

**Ingeniería en Computación  
Inteligente**

*5to Semestre*

# Índice

<b>Investigación del Videojuego</b>	<b>2</b>
Contexto y Genero del Videojuego	2
Ejemplos de este tipo de género.	3
Elementos y Mecánicas Clave	3
Características Únicas	3
Aspectos Técnicos de <i>AtropellaPollos</i>	4
¿Qué es Ursina?	4
Características Principales de Ursina	4
Contexto del Uso de Ursina en el Desarrollo de <i>AtropellaPollos</i>	5
<b>Evidencias del Desarrollo</b>	<b>6</b>
Selección del Kart	6
Desarrollo del Desplazamiento del Kart en un Plano	6
Selección y Diseño del Entorno	8
Colocación de los Pollos	9
Diseño de las Interfaces del Juego	10
<b>Requerimientos para su ejecución</b>	<b>13</b>
<b>Conclusiones</b>	<b>14</b>
Por equipo	14
Juan Francisco Gallo Ramírez	14
Pablo Emilio Soto Parada	15
<b>Programa “AtropellaPollos”</b>	<b>16</b>
<b>Bibliografía</b>	<b>17</b>

# Investigación del Videojuego

---

En este proyecto final, se decidió desarrollar un videojuego en 3D como una forma de aprendizaje y desafío personal, aprovechando la flexibilidad y las amplias posibilidades creativas que ofrece este enfoque. La elección de un entorno tridimensional permite explorar conceptos más avanzados en diseño, programación y física, además de brindar una experiencia visual inmersiva para el jugador.

El concepto del juego está inspirado en la modalidad de juegos de arcade, donde un objeto móvil se desplaza sobre una superficie, interactuando con el mundo que lo rodea. El objetivo principal será que el jugador recoja elementos dispersos en el escenario, lo que añade una capa de dinamismo y estrategia a la mecánica central. Este enfoque no solo hace el juego entretenido, sino que también plantea retos técnicos interesantes, como la implementación de físicas, detección de colisiones y gestión de objetos en tiempo real.

## **1. Contexto y Género del Videojuego.**

Nuestro videojuego *AtropellaPollos* es un videojuego en 3D que combina elementos de acción y carreras arcade. Este género es conocido por ofrecer una jugabilidad dinámica, sencilla y altamente entretenida. En este caso, el objetivo principal es controlar un vehículo que se desplaza por una superficie, atropellando pollos dispersos en el entorno. Una vez que el jugador logra atropellar a todos los pollos, se registra su récord, incentivando la superación personal y el perfeccionamiento de habilidades.

El concepto se apoya en una mecánica central simple pero efectiva: moverse, interactuar con el entorno y cumplir objetivos. Este enfoque asegura una experiencia accesible para jugadores de todas las edades, mientras que su temática humorística le otorga un carácter único y atractivo dentro de su género.

## 2. Ejemplos de este tipo de género.

*AtropellaPollos* toma elementos clave de juegos icónicos que destacan en sus respectivos géneros:

- **Mario Kart (Nintendo):** Referente de los juegos de carreras arcade, donde el manejo de vehículos es dinámico y accesible. De este juego se toma la inspiración para ofrecer controles intuitivos y una experiencia de manejo divertida.
- **Crossy Road:** Aunque no es un juego de karts, utiliza la idea de interacción simple y humorística con personajes, además de un diseño visual llamativo y accesible.
- **Crazy Taxi:** Un clásico donde el objetivo es recoger pasajeros en un entorno dinámico. De este juego se toma la idea de objetivos rápidos y rejugabilidad.

## 3. Elementos y Mecánicas Clave

- **Controles Intuitivos:** El jugador podrá acelerar, frenar y girar de manera sencilla, asegurando una experiencia de manejo fluida.
- **Interacción Dinámica:** Los pollos dispersos en el entorno actúan como los principales objetivos. Cada vez que el jugador los atropella, se genera un impacto visual y sonoro que refuerza la diversión.
- **Sistema de Progresión:** El registro de récords motiva al jugador a mejorar su rendimiento en cada partida, incentivando la repetición.

## 4. Características Únicas

El humor característico de atropellar pollos es el núcleo distintivo de *AtropellaPollos*, ofreciendo una experiencia ligera y divertida. Su diseño minimalista, tanto en mecánicas como en objetivos, lo convierte en un juego accesible y potencialmente adictivo, ideal para sesiones rápidas de entretenimiento.

Al combinar inspiración de títulos exitosos con su propia identidad humorística y dinámica sencilla, *AtropellaPollos* promete ser una propuesta creativa y atractiva, destacando por su capacidad de enganchar y divertir a los jugadores.

## 5. Aspectos Técnicos de *AtropellaPollos*

Para el desarrollo de *AtropellaPollos*, se eligió Ursina como motor de juego. Este motor, basado en Python, es una excelente opción para crear juegos 3D de manera rápida y sencilla, con un enfoque en la accesibilidad y facilidad de uso. A diferencia de motores más complejos, como Unity o Unreal Engine, Ursina está diseñado para ser más directo y menos intimidante, permitiendo que desarrolladores con conocimientos básicos de Python puedan crear experiencias de juego en 3D sin complicaciones.

## 6. ¿Qué es Ursina?

Ursina es un motor de desarrollo de juegos en 3D basado en Python, diseñado para facilitar la creación de juegos sin la complejidad de otros motores más grandes. Es un motor ligero y accesible que permite a los desarrolladores crear juegos en 3D de manera sencilla, con una curva de aprendizaje baja, lo que lo hace ideal para quienes están comenzando a programar o desean crear prototipos rápidos.

El motor Ursina se basa en Pygame, otro popular motor de Python, pero amplía sus capacidades al ofrecer herramientas y funcionalidades específicas para trabajar en 3D. Ursina simplifica el proceso de creación de juegos, desde la configuración de escenas hasta la implementación de interacciones, físicas y gráficos, todo dentro de un entorno Python.

## 7. Características Principales de Ursina

- **Simplicidad y Accesibilidad:** Ursina está diseñado para ser fácil de usar, incluso para aquellos que no tienen experiencia previa en desarrollo de videojuegos. Su sintaxis es clara y directa, lo que permite que los

desarrolladores puedan concentrarse más en la creatividad y en las mecánicas de juego.

- **Motor 3D Integrado:** A diferencia de muchos motores de juego más complejos, Ursina incluye soporte nativo para gráficos 3D, lo que permite crear juegos con un entorno tridimensional sin complicaciones adicionales.
- **Físicas Simples:** Ursina incluye un sistema de físicas básico que permite simular el movimiento de objetos, colisiones y gravedad, lo que es ideal para juegos donde estas interacciones son esenciales, como en *AtropellaPollos*.
- **Modularidad:** El motor es altamente modular, lo que significa que se pueden agregar o quitar funcionalidades fácilmente, como sonido, texturas, efectos visuales, y más. Esto da a los desarrolladores la flexibilidad para ajustar el motor según las necesidades de su proyecto.
- **Enfoque en la Productividad:** Ursina está optimizado para crear juegos rápidamente. Gracias a sus funciones integradas y su bajo umbral de entrada, los desarrolladores pueden centrarse en la creación de contenido y mecánicas, sin tener que lidiar con configuraciones complicadas o extensas.

## 8. Contexto del Uso de Ursina en el Desarrollo de *AtropellaPollos*

Dado su enfoque en la simplicidad y accesibilidad, Ursina es una opción excelente para proyectos como *AtropellaPollos*, que busca ofrecer una experiencia de juego en 3D sencilla y directa.

El motor es especialmente adecuado para juegos que no requieren gráficos extremadamente complejos o simulaciones físicas avanzadas, lo que hace que se ajuste perfectamente a la naturaleza simple y humorística de *AtropellaPollos*, un juego de acción donde el objetivo es atropellar pollos en un entorno 3D.

En resumen, Ursina es una herramienta poderosa pero accesible que permite a los desarrolladores crear juegos 3D de manera rápida y eficiente, sin la complejidad que podría implicar el uso de motores más grandes como Unity o Unreal Engine. Es el motor ideal para *AtropellaPollos*, ya que permite un enfoque ágil y flexible para lograr una experiencia de juego divertida y entretenida.

# Evidencias del Desarrollo

El desarrollo de *AtropellaPollos* siguió un enfoque estructurado y progresivo, abordando cada aspecto técnico y visual de manera organizada. A continuación, se describe en detalle cada paso del proceso:

## 1. Selección del Kart

**Objetivo:** Determinar el vehículo principal que el jugador controlará durante el juego.

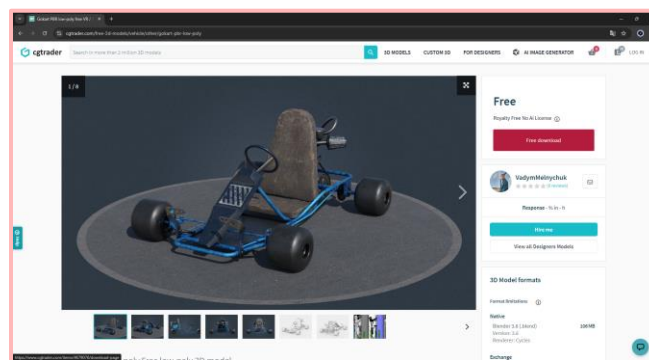
### Acciones Realizadas:

Se investigaron recursos en línea (bibliotecas 3D gratuitas y de pago) para identificar un modelo de kart que cumpliera con los requisitos del proyecto:

- Estética humorística: Compatible con el tono ligero del juego.
- Tamaño adecuado: Proporcional al entorno y a los demás elementos interactivos (como los pollos).

Se realizaron pruebas para verificar que el modelo seleccionado fuera compatible con Ursina, asegurando una importación exitosa y un rendimiento óptimo.

Se definió el color y los materiales del kart, manteniendo un diseño atractivo pero simple para garantizar buen rendimiento gráfico.



## 2. Desarrollo del Desplazamiento del Kart en un Plano

**Objetivo:** Implementar una mecánica de movimiento que se sintiera fluida, responsiva y divertida para el jugador.

## Acciones Realizadas:

### ■ Programación inicial de controles:

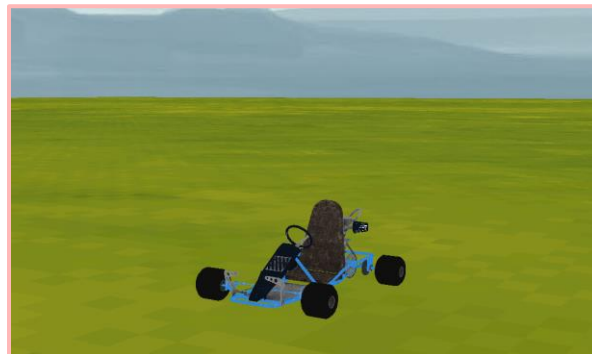
- Uso de las teclas de flecha y/o WASD para controlar el movimiento.
- Configuración de acciones como acelerar (W), frenar (S) y girar (A/D).
- Posicionamiento respectivo de la cámara en el kart.

### ■ Implementación de físicas:

- Se simuló el comportamiento de un vehículo realista pero simplificado, utilizando velocidades progresivas y frenado gradual, con efecto de sonido.
- Se definió una fricción básica para evitar que el kart se deslice demasiado en el plano.

### ■ Pruebas iterativas:

- Se ajustaron variables como velocidad máxima, aceleración y radios de giro para lograr una experiencia satisfactoria.





```

#===== COMPORTAMIENTO DEL KART =====
def kartBehavior():
    global kart
    global speed, max_speed, max_speed_reverse
    global acceleration, acceleration_reverse, friction
    global direction_speed, direction_proportion
    global hit
    global brake
    global started

    #===== Velocidad del vector de movimiento =====
    acc = 0

    if held_keys['w'] and held_keys['s']:
        if speed > 0:
            acc = acceleration_reverse
        elif speed < 0:
            acc = acceleration
        elif speed == 0:
            acc = 0
    elif held_keys['w']:
        acc = acceleration
        if not kart_sound.playing:
            kart_sound.play()
        brake = False
    elif held_keys['s']:
        acc = acceleration_reverse

        if not kart_sound.playing:
            kart_sound.play()
        if speed > 0 and not brake and not kart_brake.playing:
            kart_brake.play()
            brake = True

    if speed != 0 and not started:
        started = True
        showControls()

    if speed > 0 or speed == 0 and acc > 0:
        speed = clamp(speed + (acc + friction) * time.dt,
                      0,
                      max_speed)
    elif speed < 0 or speed == 0 and acc < 0:
        speed = clamp(speed + (acc - friction) * time.dt,
                      max_speed_reverse,
                      0)

```

```

#===== Dirección del vector de movimiento =====
if held_keys['a'] and speed != 0:
    kart.rotation_y += (1 if speed > 0 else -1) * (max_speed - direction_proportion * abs(speed)) * -direction_speed * time.dt
if held_keys['d'] and speed != 0:
    kart.rotation_y += (1 if speed > 0 else -1) * (max_speed - direction_proportion * abs(speed)) * direction_speed * time.dt

#===== Concretar posición del objeto =====
if kart.intersects().hit:
    #----- Mecánica de golpe.
    speed = 0
    kart.z *= 0.999
    kart.x *= 0.999

    if not hit and not hit_sound.playing:
        hit_sound.play()
        hit = True
else:
    #----- Movimiento normal.
    kart.z += math.cos(math.radians(kart.rotation_y)) * speed
    kart.x += math.sin(math.radians(kart.rotation_y)) * speed
    hit = False

if not kart_sound.playing and not kart_engine.playing:
    kart_engine.play()

```

### 3. Selección y Diseño del Entorno (Árboles, Plantas y Rocas)

**Objetivo:** Crear un entorno tridimensional que fuera atractivo visualmente y complementara la jugabilidad.

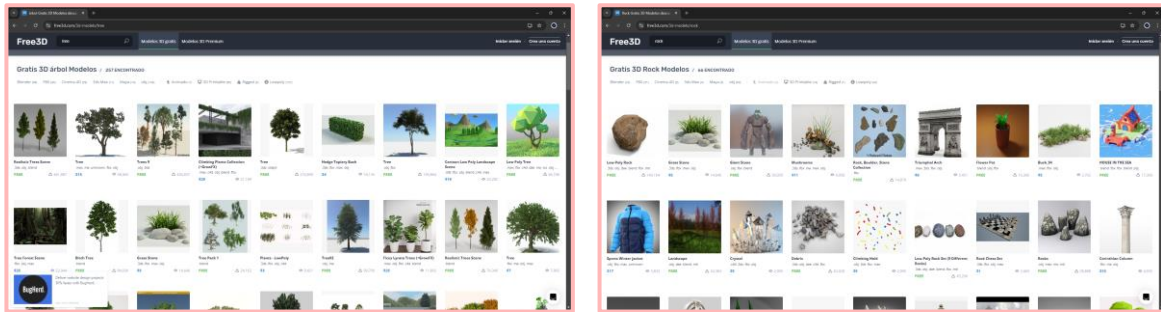
#### Acciones Realizadas:

##### ▪ Selección de elementos del entorno:

- Se eligieron modelos simples de muros, árboles, plantas y rocas disponibles en bibliotecas públicas de recursos 3D.

- **Colocación estratégica:**

- Los elementos fueron distribuidos en el plano para evitar que bloquearan las rutas principales del kart, pero al mismo tiempo ofrecer una sensación de paisaje variado.
- Se generaron agrupaciones de elementos (por ejemplo, un grupo de árboles o una formación rocosa) para dar más naturalidad al escenario.



#### 4. Colocación de los Pollos

**Objetivo:** Integrar los objetivos principales del juego (los pollos) en el entorno.

**Acciones Realizadas:**

- **Diseño de los pollos:**

- Modelos simples en 3D con un diseño amigable y caricaturesco.

▪ **Programación de interacción:**

- Se implementó una detección de colisión entre el kart y los pollos para registrar un "atropello".
- Se añadieron efectos visuales y sonoros al atropellar un pollo para mejorar la retroalimentación al jugador.
- Distribución en el mapa:
  - *Los pollos fueron colocados en lugares estratégicos, incentivando al jugador a explorar todo el escenario.*
  - *Se implementaron diferentes patrones de aparición, como pollos agrupados en zonas o distribuidos de manera uniforme.*



## 5. Diseño de las Interfaces del Juego

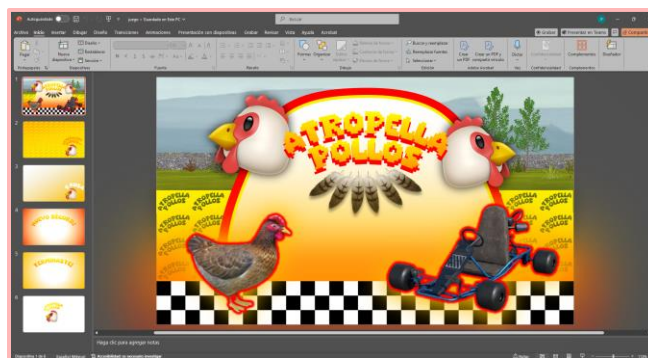
**Objetivo:** Crear menús y HUD (Heads-Up Display) que sean intuitivos y mejoren la experiencia del usuario.

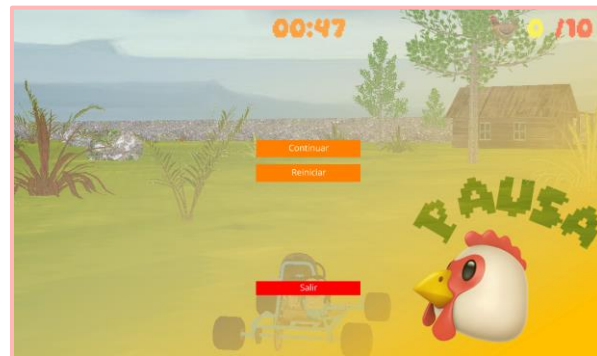
**Acciones Realizadas:**

▪ **Menú principal:**

- Se diseñó una pantalla de inicio simple con opciones como "Iniciar Juego" y "Salir".

- Se añadió un fondo animado para mantener una estética coherente con el juego.
- **HUD:**
  - Se programó una barra de progreso que muestra cuántos pollos han sido atropellados y cuántos faltan.
  - Se incluyó un contador de tiempo para registrar el récord del jugador.
- **Pantalla de fin de juego:**
  - Una pantalla que aparece al finalizar la partida, mostrando el puntaje obtenido y el tiempo logrado.
  - Botones para reiniciar el juego o volver al menú principal.





# Requerimientos para su ejecución

Para ejecutar el videojuego, evidentemente es necesario tener Python instalado, sin embargo también requiere tener instalada la librería del motor gráfico de Ursina. Se puede instalar escribiendo en la terminal (con Python instalado previamente):

```
pip install ursina
```

Una vez instalada la librería, debería poder ejecutarse sin ningún problema, recalcando que el programa debe estar en su carpeta fuente, ya que ahí se encuentran todos los recursos que se utilizan para generar el juego (modelos 3d, imágenes y audios).

Entidades	01/12/2024 04:10 p. m.	Carpeta de archivos	
Fonts	02/12/2024 07:18 p. m.	Carpeta de archivos	
Imagenes	04/12/2024 11:09 p. m.	Carpeta de archivos	
models_compressed	01/12/2024 03:43 a. m.	Carpeta de archivos	
Records	04/12/2024 09:49 p. m.	Carpeta de archivos	
Sonidos	03/12/2024 12:49 a. m.	Carpeta de archivos	
Texturas	01/12/2024 04:53 p. m.	Carpeta de archivos	
AtropellaPollos	05/12/2024 02:17 a. m.	Archivo de origen ...	34 KB



# Conclusiones

---

## Conclusiones (Por equipo):

El desarrollo de Atropellapollos representa un logro significativo en términos de aprendizaje, creatividad y aplicación de conocimientos en desarrollo de videojuegos. Desde el inicio, el proyecto fue concebido con la intención de explorar nuevas habilidades, aceptar retos técnicos y, al mismo tiempo, crear una experiencia de juego sencilla y entretenida.

El uso del motor Ursina fue un acierto clave, ya que permitió implementar un juego en 3D de manera eficiente y accesible, sin una curva de aprendizaje excesivamente compleja. A través del proceso, se adquirieron habilidades esenciales como la programación de físicas básicas, la creación de un entorno visual atractivo y la integración de elementos interactivos que enriquecen la jugabilidad. Además, se logró desarrollar interfaces intuitivas que complementan la experiencia del jugador, ofreciendo claridad y dinamismo en cada partida.

Más allá de lo técnico, el proyecto demuestra cómo una idea simple puede convertirse en una experiencia de juego divertida y efectiva. La elección de una temática humorística y una mecánica directa pero envolvente refleja el valor de la creatividad en el desarrollo de videojuegos. Atropellapollos no solo es un videojuego funcional, sino también una prueba del crecimiento en habilidades y conocimientos que este tipo de proyectos puede ofrecer. Este trabajo abre la puerta a nuevos retos en el mundo del desarrollo de videojuegos y marca el inicio de un camino con infinitas posibilidades creativas.

## Conclusiones (Individuales):

### ● Juan Francisco Gallo Ramírez

Trabajar en AtropellaPollos fue una experiencia enriquecedora que disfruté enormemente. La programación en Python, combinada con el uso del motor Ursina, resultó ser una combinación interesante y accesible. Ursina, con su simplicidad y flexibilidad, me permitió enfocarme más en la lógica del juego y menos en las complicaciones técnicas, lo cual hizo que el desarrollo fuera más fluido y entretenido.

La idea de crear un juego tipo arcade fue especialmente atractiva porque su estructura sencilla facilita mucho el proceso de programación. Este tipo de juegos permite centrarse en la jugabilidad y en los pequeños detalles que hacen divertida la experiencia del jugador. Además, trabajar en un concepto directo como atropellar pollos en un kart me permitió explorar la creatividad y experimentar con diferentes mecánicas, todo en un ambiente relajado y lleno de aprendizaje.

Este proyecto no solo me ayudó a mejorar mis habilidades técnicas, sino que también confirmó lo mucho que disfruto el proceso de dar vida a una idea a través de la programación. Es una experiencia que sin duda quisiera repetir y expandir en futuros proyectos.

#### ● **Pablo Emilio Soto Parada**

Trabajar en AtropellaPollos fue una experiencia fascinante, especialmente porque me permitió implementar fenómenos físicos como el desplazamiento del kart. Crear un sistema de movimiento que se sintiera fluido y natural fue uno de los aspectos más emocionantes del proyecto. Ajustar la aceleración, el frenado, los giros y la interacción con el entorno me hizo apreciar la complejidad y belleza de la física en los videojuegos.

El proceso de traducir conceptos físicos al código fue un desafío, pero también muy satisfactorio. Cada ajuste que lograba mejorar el comportamiento del kart hacía que el juego se sintiera más realista y divertido. Esto no solo mejoró mi comprensión de las mecánicas físicas, sino que también reforzó mi interés por explorar más este aspecto en futuros proyectos. Sin duda, fue una experiencia enriquecedora que me motivó a seguir aprendiendo y experimentando en el desarrollo de videojuegos.



# Programa “AtropellaPollos”

---

Debido a que el peso de la carpeta del programa es mayor al límite establecido para la entrega en la plataforma de AulaVirtual se anexa un enlace que lleva a la carpeta comprimida en OneDrive:

- [AtropellaPollos.zip](#)

Sin embargo, en la entrega se anexa de todas formas el archivo Python que contiene toda la lógica del programa, con la limitante de que no se puede ejecutar sin los recursos de la carpeta del programa.

# Bibliografía

---

Amland, P. (s.f.). *ursinaengine.org*. Obtenido de ursina engine:  
<https://www.ursinaengine.org/index.html>

Python Software Foundation. (s.f.). *Python.org*. Obtenido de The Python  
Language Reference: <https://docs.python.org/3/reference/index.html>

Wikipedia. (s.f.). *Wikipedia*. Obtenido de Arcade:  
<https://es.wikipedia.org/wiki/Arcade>