



Universidad
Gerardo Barrios

Facultad:

Ciencia y Tecnología.

Carrera:

Ingeniería en sistemas y redes informáticas.

Materia:

Programación Computacional III.

Tema:

Parcial II Computo II.

Docente:

Willian Alexis Montes Girón.

Integrantes:

Diego Martín López Moreno (SMSS097824)

Walter Alexander Ramírez Benítez (SMSS082124)

Juan Ramón Espinal Coto (SMSS102323)

Franklin Aldahir Portillo Flores (SMSS011624)

Fecha: 21, Octubre 2025.

Índice

No. Pág

Introducción.....	3
¿En qué consiste la librería VPython y para qué se usa?.....	4
Usos principales de VPython	4
Funciones más relevantes y utilizadas de la librería.	5
1. sphere() – Crear esferas 3D (ampliada).....	5
2.Box() – Crear cubos o bloques.	6
3. cylinder() – Creación de cilindros tridimensionales.....	7
4. Arrow() – Representación de vectores en 3D.....	8
5. Rotate() – Rotar objetos en el espacio 3D.	9
Casos de uso en los que VPython se ha utilizado en la vida real	10
____ Educación	10
____ Proyecto Phythones - Simulaciones Físicas Tridimensionales	10
____ Mecánica Celeste y Astronomía	11
____ Simulación de Movimiento de Projectiles.....	12
____ Investigación en Incendios (Fire Research)	13
Conclusión.....	14
Bibliografías	15

Introducción.

En la actualidad, Python se ha consolidado como uno de los lenguajes de programación más versátiles y utilizados, gracias a su sintaxis sencilla y a la gran cantidad de librerías externas que facilitan el desarrollo de proyectos en distintas áreas. Entre las librerías más llamativas e interactivas se encuentra VPython, una herramienta diseñada para la creación de gráficos tridimensionales (3D) y simulaciones visuales en tiempo real. Python por sí solo no ofrece salida gráfica. La biblioteca de gráficos Tk permite crear gráficos 2D, pero está dirigida a programadores con mucha experiencia y no es compatible con visualizaciones 3D.

VPython permite a los usuarios generar escenas y objetos en tres dimensiones con unas pocas líneas de código, ofreciendo una experiencia visual atractiva y didáctica. Esto la convierte en una opción popular tanto en el ámbito educativo para enseñar conceptos de física, matemáticas y programación como en proyectos científicos o experimentales que requieren visualización dinámica.

Sin embargo, esta librería surgió para lograr crear un módulo de gráficos 3D para Python, llamado "Visual", extremadamente fácil de usar. Un programa puede crear objetos 3D (como esferas, curvas, etc.) y posicionarlos en el espacio 3D. Visual, ejecutándose en un hilo independiente, actualiza automáticamente una escena 3D varias veces por segundo para reflejar la posición actual de los objetos.

El programador no necesita gestionar la visualización, sino que puede centrarse en los aspectos computacionales del programa. El usuario puede navegar por la escena 3D usando el ratón para acercar y girar mientras el programa se ejecuta. Visual es totalmente compatible con álgebra vectorial. La combinación de Python y el módulo Visual se denomina "VPython". Visual también depende de un módulo de matriz numérica: numpy.

VPython obtiene mucha demanda en diferentes áreas como:

Uso en la educación

Los estudiantes de cursos introductorios de física han estado usando VPython para modelar por computadora. VPython les permite centrarse en los cálculos físicos sin tener que escribir instrucciones gráficas explícitas, y aun así obtener visualización 3D. Pueden realizar cálculos vectoriales reales, lo que mejora su comprensión de la utilidad de los vectores y la notación vectorial. Los instructores han usado VPython para crear programas de demostración en 3D para sus clases.

Uso en investigación

Python se usa ampliamente en la investigación científica (véase scipy.org). VPython ha resultado útil en la investigación para visualizar situaciones y datos en 3D.

Su principal fortaleza radica en la simplicidad con la que se pueden crear modelos tridimensionales interactivos sin la necesidad de conocimientos avanzados en gráficos por computadora. Además, su integración con GlowScript permite ejecutar simulaciones directamente desde el navegador, facilitando la demostración y el intercambio de proyectos sin necesidad de instalaciones complejas.

¿En qué consiste la librería VPython y para qué se usa?

VPython es una librería de Python diseñada para la creación y visualización de gráficos tridimensionales (3D) de manera sencilla y rápida. Su principal objetivo es permitir a los programadores, estudiantes e investigadores representar objetos y simulaciones en un espacio tridimensional sin necesidad de conocimientos avanzados de gráficos por computadora o motores 3D complejos.

Esta librería consiste en generar objetos básicos como esferas, cubos, cilindros, flechas y planos, así como la posibilidad de aplicar colores, texturas, animaciones y trayectorias a dichos objetos. Además, VPython incluye herramientas para manipular coordenadas tridimensionales mediante vectores, controlar la velocidad de animación y crear interacciones en tiempo real. Permite la creación de objetos tridimensionales y su animación en tiempo real con un mínimo esfuerzo de programación. Su simplicidad la hace ideal para personas que inician en programación 3D, ya que no requiere conocimientos de motores gráficos complejos como OpenGL o Unity.

VPython facilita la creación de pantallas y animaciones 3D navegables, incluso para quienes tienen poca experiencia en programación. Al estar basado en Python, también ofrece mucho que ofrecer a programadores e investigadores con experiencia.

Usos principales de VPython

Educación y aprendizaje de conceptos científicos:

Enseña de manera visual fenómenos de física y matemáticas, como el movimiento de proyectiles, trayectorias, fuerzas, órbitas planetarias o péndulos.

Facilita la comprensión de conceptos abstractos al traducir fórmulas matemáticas en representaciones visuales interactivas.

Simulación de fenómenos y procesos:

Representa dinámicas de sistemas mecánicos, estructuras y movimientos de partículas.

Permite experimentar con variables en tiempo real y observar cómo afectan el comportamiento de los objetos.

Prototipos visuales y demostraciones científicas:

Se utiliza para crear modelos tridimensionales de moléculas, estructuras físicas o diseños de ingeniería antes de la implementación real.

Proyectos creativos y artísticos:

Algunos desarrolladores y educadores crean animaciones interactivas o visualizaciones artísticas usando VPython como herramienta de presentación.

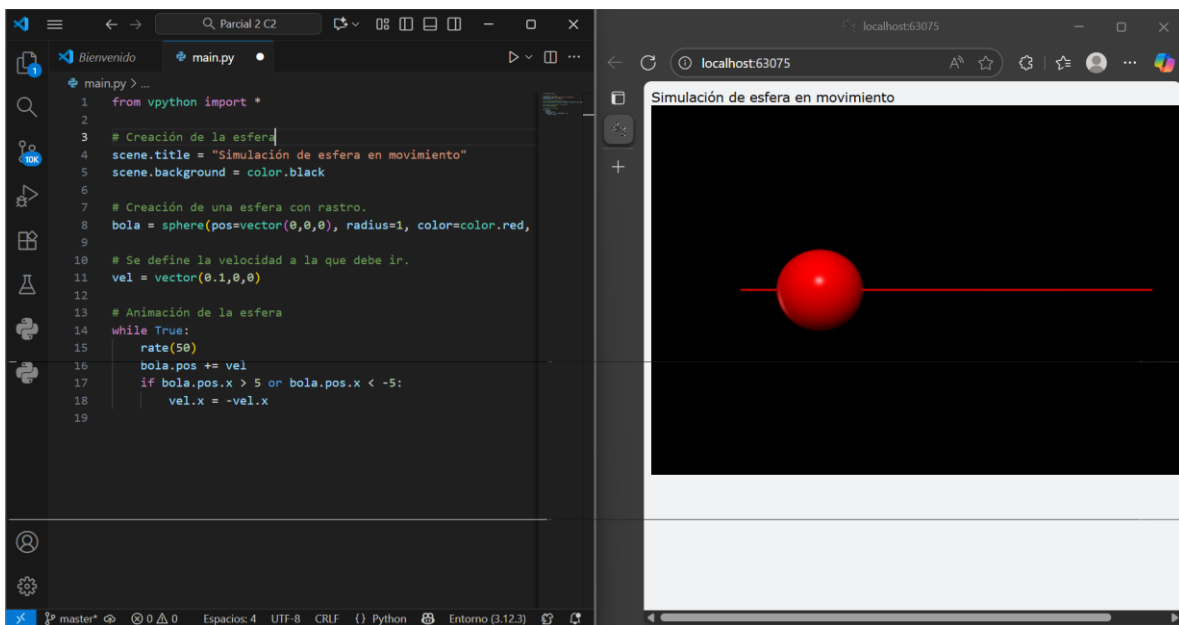
Funciones más relevantes y utilizadas de la librería.

VPython se basa en objetos 3D y funciones que permiten crear y animar estos objetos de manera sencilla. A continuación, se presentaran funciones o clases fundamentales.

1. sphere() – Crear esferas 3D (ampliada)

La función sphere() permite crear esferas tridimensionales en la escena de VPython. Esta función es una de las más importantes porque la esfera es un objeto universal en simulaciones físicas y educativas. Puede representar planetas, partículas, pelotas, moléculas o cualquier objeto redondeado, dependiendo del contexto de la simulación.

En física, la función sphere() se utiliza para enseñar conceptos de movimiento rectilíneo uniforme, acelerado o proyectiles. Por ejemplo, al programar una esfera que cae bajo la influencia de la gravedad, los estudiantes pueden observar cómo varía la velocidad y la posición de un objeto con el tiempo. Esto ayuda a que los conceptos teóricos se vuelvan visualmente comprensibles, especialmente para alumnos que aprenden mejor mediante representaciones gráficas.



La función sphere() genera un objeto con propiedades visuales como tamaño, color y posición.

make_trail=True permite que la esfera deje un rastro de movimiento, útil para ver trayectorias.

rate(50) controla la velocidad de la animación para que sea fluida y observable.

El bucle while simula el movimiento continuo de la esfera, incluyendo el rebote en los límites.

2.Box() – Crear cubos o bloques.

La función `box()` permite crear objetos rectangulares o cúbicos que representan suelos, paredes, plataformas o bloques constructivos. Es fundamental para dar contexto o referencia espacial a los objetos en movimiento. Por ejemplo, una esfera puede rebotar sobre un piso generado por un `box()`, haciendo la simulación más realista.

La función `box()` genera prismas rectangulares (cuboides) en el espacio tridimensional. Estos objetos se utilizan como pisos, muros, plataformas o contenedores, y proporcionan referencia espacial para otros objetos que interactúan con ellos.

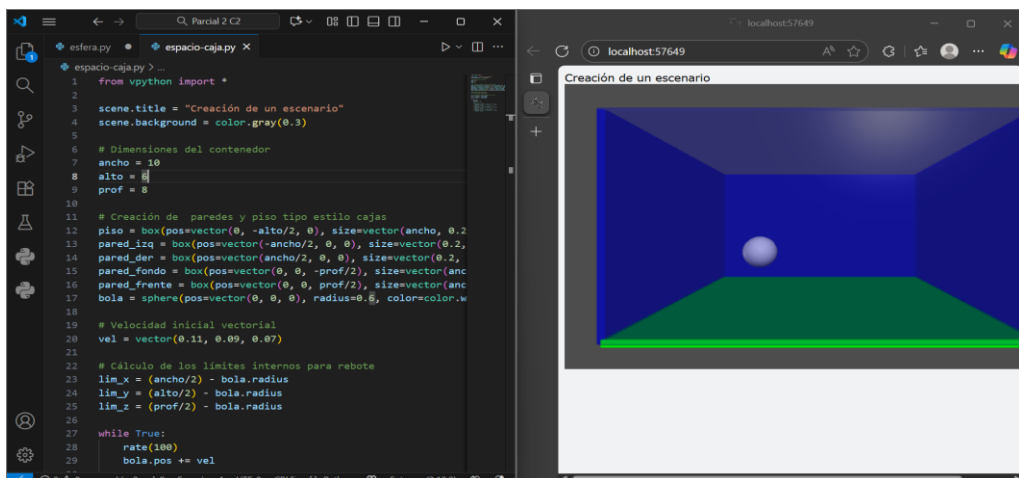
Parámetros más comunes:

`pos` — posición del centro del cubo.

`size (vector)` — tamaño del cubo en los tres ejes (largo, alto, profundidad).

`color` — color del objeto.

`opacity (opcional)` — transparencia del objeto (entre 0 y 1), útil para ver lo que hay dentro o detrás.



Mediante varias llamadas a `box()` se define un contenedor tridimensional: piso, paredes izquierda y derecha, fondo y frente.

La esfera se ubica en el centro con radio 0.6, y el contenedor es suficientemente grande para que la esfera pueda moverse libremente.

Se calcula el límite interno para rebote restando el radio de la esfera al tamaño del contenedor en cada eje.

La esfera rebota al tocar cualquier pared o el piso, también el techo invirtiendo la componente de velocidad correspondiente.

`make_trail=True` permite ver claramente el recorrido 3D.

3. cylinder() – Creación de cilindros tridimensionales.

La función `cylinder()` permite crear objetos cilíndricos en el espacio 3D, útiles para representar ejes, varillas, tubos o componentes estructurales en simulaciones físicas o mecánicas.

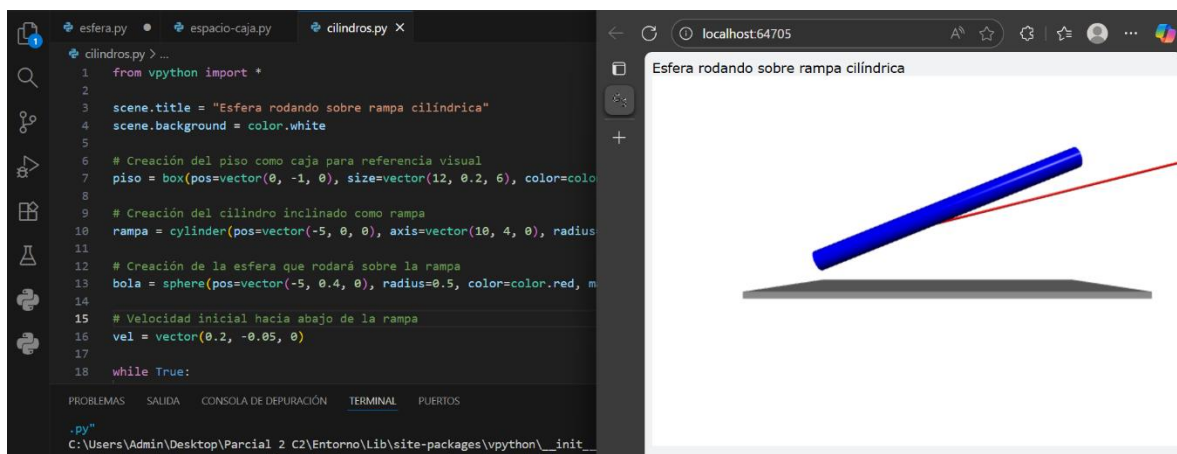
Parámetros principales:

`pos` — punto de inicio del cilindro (base).

`axis (vector)` — define la dirección y la longitud del cilindro.

`radius` — radio del cilindro (grosor).

`color` — color del cilindro.



Se crea un cilindro inclinado (rampa) con punto de arranque en $(-5, 0, 0)$ y un eje $(10, 4, 0)$, lo que le da pendiente.

La esfera se ubica justo encima del cilindro con una separación vertical igual al radio.

La velocidad inicial tiene componente paralela a la rampa; al bajar, la esfera “rebota” ligeramente si toca la rampa, invirtiendo su componente vertical.

En simulaciones reales se podrían usar normales y proyecciones para simular el contacto con la rampa de forma física (fuerza normal, fricción).

Este tipo de visualización permite ilustrar cómo un objeto rueda o interactúa con una superficie inclinada.

Muy útil para mostrar **movimiento sobre planos inclinados**, fuerzas normales, componentes del peso o proyecciones vectoriales.

4. Arrow() – Representación de vectores en 3D

La función `arrow()` crea flechas tridimensionales que se usan para representar vectores como velocidad, fuerza, aceleración o dirección. Es especialmente útil en entornos educativos para visualizar conceptos vectoriales que muchas veces son abstractos en clases teóricas.

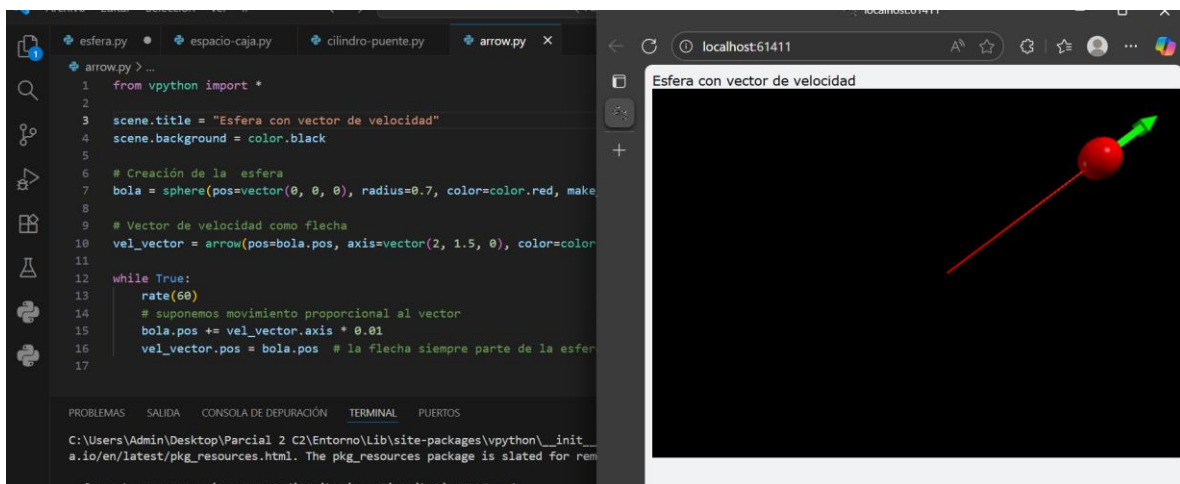
Parámetros más comunes:

`pos` — punto de origen de la flecha.

`axis (vector)` — dirección y magnitud de la flecha.

`color` — color de la flecha (por ejemplo `color.cyan`, `color.orange`).

`shaftwidth`, `headwidth`, `headlength` (opcional) — permiten ajustar grosor y tamaño de cabeza de flecha.



Explicación del ejemplo:

Se crea una esfera roja con rastro.

Se define una flecha verde que apunta en la dirección del vector `axis`.

En cada iteración, la esfera se mueve proporcionalmente a la flecha, y la flecha se reposiciona para originarse en la esfera.

Visualmente muestra cómo la dirección y magnitud del vector influyen en el movimiento.

Permite que se vea directamente vectores en el espacio 3D, algo que de otro modo se hace con diagramas 2D.

Muy útil cuando se combina con `sphere()` u otros objetos para mostrar interacción entre objetos y vectores.

5. Rotate() – Rotar objetos en el espacio 3D.

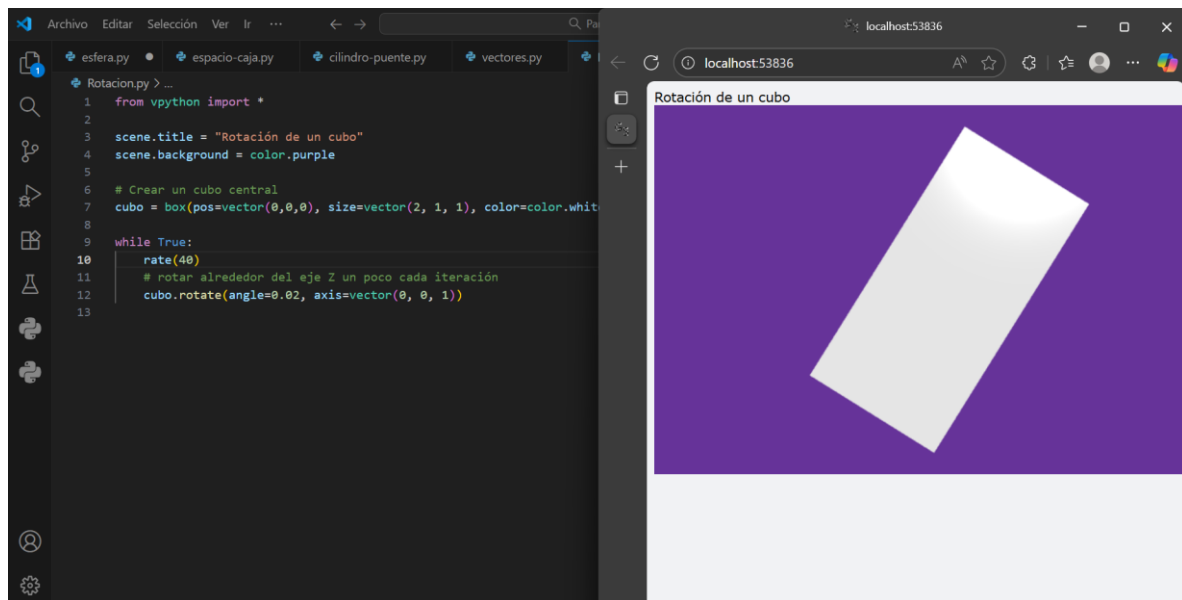
Rotate() es esencial para manipular objetos en VPython en términos de orientación o rotación. Esto permite hacer simulaciones más dinámicas y realistas.

Parámetros importantes:

angle — ángulo (en radianes) de rotación.

axis (vector) — eje alrededor del cual rotar (por ejemplo vector(1,0,0) para rotar alrededor del eje x).

origin (opcional) — punto alrededor del cual se realiza la rotación. Por defecto es el origen del objeto.



Se genera un cubo de dimensiones $2 \times 1 \times 1$ en el centro.

Dentro del bucle, en cada fotograma se aplica una rotación pequeña en el eje Z (axis=vector(0,0,1)).

El cubo va girando lentamente sobre sí mismo, mostrando su estructura desde distintos ángulos.

Puedes combinar rotaciones sobre diferentes ejes (X, Y, Z) para efectos más complejos.

La función rotate() en VPython resulta fundamental cuando se busca dar realismo o dinamismo a una escena tridimensional. Gracias a ella, los objetos pueden girar alrededor de un eje determinado, simulando comportamientos físicos o mecánicos de la vida real, como el movimiento de un planeta alrededor de su estrella, la rotación de una hélice o el giro de una rueda. Su aplicación no se limita únicamente a movimientos estéticos, sino que también permite modelar sistemas donde la orientación de los cuerpos es un factor determinante, como en simulaciones de ingeniería, robótica o astronomía.

Casos de uso en los que VPython se ha utilizado en la vida real

Educación

Actualmente, el mundo de la física se enfrenta a la incompreensión de las personas, debido a que para muchos es difícil entender la belleza de esta ciencia; pero la informática ha llegado para dar una luz de entendimiento, con programas y herramientas visuales que han comenzado a apoyar fuertemente en su enseñanza.

El incremento en el poder computacional ha permitido que la academia desarrolle la educación en física búsqueda computacional la integración De la warela- nformática estafa la física, para obtener mejores resultados en el aprendizaje que la forma habitual de dictar los cursos.

La educación en ciencias computacionales presenta l Naciones Unidas camino para seguir, pues integra las experiencias para estimular y avaler a los estudiantes de la ingeniería el deseo de generar una base sólida del conocimiento, útil para su experiencia en relaciones públicas profesionales.

En la Universidad de los Llanos, estudiantes de Ingeniería Electrónica y Ciencias de la Computación utilizaron VPython como parte de las actividades del grupo de investigación Sistemas Dinámicos para el aprendizaje de física computacional mediante cinemática de partículas.

Las librerías empleadas (matplotlib, numpy, PyQt4, scipy, Tkinter y VPython) permitieron la simulación de movimiento uniforme, movimiento lineal acelerado, caída libre y movimiento de proyectiles, además de ser útiles para la generación de interfaces gráficas de usuario para mostrar datos en tablas y gráficos

Proyecto Phythones - Simulaciones Físicas Tridimensionales

La visualización y creación de escenas dinámicas tridimensionales (y no solo el ploteando de funciones) pueden tener un gran valor didáctico en muchos aspectos de la física básica como la dinámica del sólido rígido, la dinámica de fluidos o el electromagnetismo. El Proyecto Phythones es un proyecto libre para construir una colección amplia de simulaciones físicas tridimensionales en Visual Python, siguiendo el paradigma del software libre.

Mecánica Celeste y Astronomía

La mecánica celeste es una rama de las ciencias que se encarga de estudiar el movimiento de los cuerpos celestes. Fue desarrollado a lo largo del siglo xvii por el estudio y la observación del sistema solar. En los siglos XVII y XVIII, cuando se hablaba de astronomía se hablaba de mecánica celeste, gracias a los trabajos observacionales de Kepler, el soporte teórico de la teoría gravitacional de Newton y los trabajos de Laplace y Lagrange. El desarrollo de la mecánica celeste, apoyado en la tecnología y los métodos computacionales propició que los países desarrollados generaran proyectos que permitieron poner en órbita diferentes satélites en las décadas de los cincuenta y sesenta, los cuales al ser optimizados se convirtieron en la herramienta fundamental de las telecomunicaciones, el posicionamiento global y la cartografía, entre otros.

El lenguaje de programación VPython es una plataforma didáctica bastante efectiva para los usuarios por su lenguaje orientado a objetos, su entorno de simulación y animación. Diferentes trabajos en el ámbito de las ciencias, y que han sido reconocidos en espacios de divulgación, han sido llevados a cabo en este lenguaje de programación. Uno de ellos fue realizado en la Universidad de Antioquia en Colombia por el profesor Jorge Zuluaga (2013), en la estimación de la posible órbita del meteorito que colisionó ese año en Chelyabinsk, Rusia. Se diseñó un programa en VPython para simular la trayectoria de planetas y naves espaciales, específicamente para enseñar transferencias orbitales de Hohmann.

El programa se diseñó para que estudiantes de cursos introductorios de astronomía visualicen las posibles trayectorias y comprendan las características principales de transferencias orbitales usando como ejemplo el viaje de una nave espacial de la Tierra a Marte y a Júpiter. El programa también permite determinar las características principales de las órbitas planetarias y relacionar los parámetros de las elipses con la energía y el momentum angular de los planetas.

VPython ha sido utilizado en trabajos reconocidos en el ámbito científico, como el realizado en la Universidad de Antioquia en Colombia por el profesor Jorge Zuluaga en 2013, en la estimación de la posible órbita del meteorito que colisionó ese año en Chelyabinsk, Rusia.

Estos programas se empezaron a utilizar como apoyo en la enseñanza del curso electivo de Astronomía General que ofrece el Departamento de Física de la Universidad Pedagógica Nacional.

Simulación de Movimiento de proyectiles

El proyecto de simulación de movimiento de proyectiles con Python surge de la necesidad de proporcionar herramientas visuales interactivas que permitan a los estudiantes comprender mejor los principios físicos involucrados en este tipo de movimiento. Las simulaciones realizadas en Python tienen la ventaja de reproducir con gran exactitud la mayoría de los fenómenos físicos que se estudian en las aulas, permitiendo que el estudiante se familiarice con el fenómeno estudiado y desarrolle sus propios modelos

Se desarrolló un código en el software libre Python que caracteriza la trayectoria de un proyectil en condiciones ideales, como herramienta para mejorar el aprendizaje del movimiento de proyectiles en cursos de física clásica para estudiantes de ciencias e ingenierías

La simulación implementada permite:

Visualización tridimensional en tiempo real: Mediante VPython, los estudiantes pueden observar la trayectoria del proyectil desde múltiples ángulos, rotando la cámara y ajustando el zoom para una mejor comprensión del movimiento.

Parametrización completa: Los usuarios pueden modificar variables como:

Velocidad inicial del proyectil

Ángulo de lanzamiento

Altura inicial

Condiciones ambientales (en versiones más avanzadas)

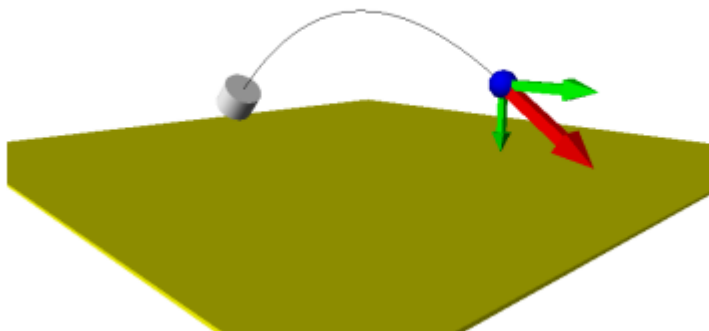
Cálculo y visualización de datos: El programa calcula automáticamente:

Alcance máximo del proyectil

Altura máxima alcanzada

Tiempo de vuelo

Coordenadas posición-tiempo



Investigación en Incendios (Fire Research)

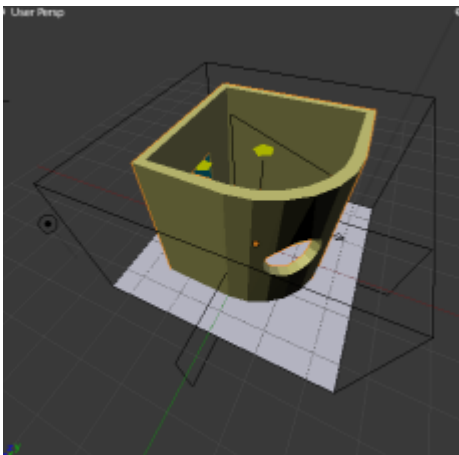
El uso de VPython en la ingeniería de protección contra incendios y la investigación sobre incendios tiene muchas aplicaciones útiles y permite una abundancia de posibilidades para la comunidad científica del fuego. Recientemente, VPython se ha utilizado para crear herramientas de análisis de datos para experimentos de incendios y datos de simulación, para ayudar a los usuarios con el modelado y la prevención de incendios. cálculos dinámicos, para realizar el seguimiento de la altura de la llama para el almacenamiento. incendios domésticos y pruebas a escala de banco utilizando programas GUI, y para resultados actuales de propagación de llamas utilizando un método visual de superposición.

Trazado de vídeo posado. Estas herramientas han surgido en forma de aplicaciones web, GUI y scripts de reducción de datos que interactúan con herramientas de dinámica de fluidos computacional (CFD) existentes, como el programa de modelado de incendios de código abierto y disponible gratuitamente, Fire Simulador de dinámica [FDS], que es mantenido por la Administración Nacional Instituto de Estándares y Tecnología (NIST).

Kristopher Overholt utilizó Python (mencionando VPython entre las librerías) en su investigación sobre incendios en el Instituto Nacional de Estándares y Tecnología (NIST) de Estados Unidos (PDF) Numerical Pyromaniacs: The Use of Python in Fire Research. El trabajo incluía el desarrollo de herramientas para la comunidad de investigación de incendios, aprovechando la naturaleza libre y de código abierto de Python y su comunidad activa.

Respecto a la creciente cantidad de interacción entre Python y modelos de fuego, desarrolladores externos en la comunidad de modelado de fuego La comunidad (incluyéndome a mí) ha lanzado recientemente una herramienta para modelar Geometría 3D y generar un archivo de entrada basado en texto para el FDS software de modelado de incendios. La herramienta se llama BlenderFDS y es una Extensión para [Blender] escrita en Python.

Antes de... Con el lanzamiento de BlenderFDS, los usuarios de FDS tuvieron que crear geometría para un caso ya sea manualmente usando un editor de texto o usando un programa comercial Interfaz de usuario. Ahora, utilizando BlenderFDS, los usuarios de FDS pueden crear edificios complejos y geometría irregular (por ejemplo, cilindros, ángulos techos) y automáticamente tienen la geometría dividida en el formato rectilíneo que requiere FDS.



Conclusión

A Pesar de sus limitaciones en contextos industriales, sería un error desestimar el valor de VPython. Su verdadera fortaleza radica en democratizar la visualización 3D y la simulación física, permitiendo que estudiantes, profesores, investigadores en formación y entusiastas de la física puedan crear simulaciones sofisticadas con una curva de aprendizaje mínima. VPython ha cumplido brillantemente su propósito educativo, permitiendo que miles de estudiantes comprendan conceptos físicos abstractos mediante visualizaciones interactivas que serían prohibitivamente complejas de crear con herramientas profesionales.

VPython ocupa un espacio importante en el espectro de herramientas de visualización científica, específicamente como:

- Punto de entrada: Una primera aproximación a la visualización 3D programática
- Herramienta de prototipado rápido: Para probar ideas conceptuales antes de implementarlas en sistemas más complejos.

Recurso pedagógico: Insustituible para la enseñanza de física y conceptos de programación visual. Para que VPython pudiera expandirse hacia aplicaciones más profesionales, necesitaría:

Modernización del motor de renderizado para mejorar el rendimiento

API mejorada de exportación para integración con otras herramienta

VPython ha encontrado su lugar en el ecosistema Python como una **puerta de entrada** accesible y efectiva al mundo de la simulación física y la visualización 3D, sirviendo como piedra angular en la educación física y como herramienta de prototipado conceptual rápido. Su legado no se mide en implementaciones industriales, sino en la cantidad de estudiantes e investigadores que han podido visualizar y comprender conceptos físicos complejos gracias a su simplicidad y accesibilidad.

Para profesionales y empresas que buscan soluciones de visualización 3D, simulación física o aplicaciones industriales, la recomendación clara es explorar las numerosas alternativas especializadas del ecosistema Python que están diseñadas específicamente para esos propósitos. Sin embargo, para educadores, estudiantes y cualquiera que desee aprender los fundamentos de la visualización científica y la simulación física con Python, VPython continúa siendo una opción incomparable y altamente recomendable.

Bibliografías

<https://www.glowscript.org/docs/VPythonDocs/index.html>

<https://vpython.org/contents/overview.html>

<http://how.dev/answers/what-is-vpython>

<https://sasa.musiclab.si/eri1/INFORMATIKA/programiranje/vpython/REFERENCE/reference.html>

https://www.researchgate.net/publication/305311111_DESARROLLO_DE_APLICACIONES_EN_PYTHON_PARA_EL_APRENDIZAJE_DE_FISICA_COMPUTACIONAL_Development_of_Python_applications_for_learning_computational_physics

file:///C:/Users/Admin/Downloads/Proyecto_Physthones_simulaciones_fisicas.pdf

<https://www.redalyc.org/journal/6142/614264960003/html/>

<https://steemit.com/stem-espanol/@germanmontero/movimiento-de-un-proyectil-con-simulacion-phyton>

file:///C:/Users/Admin/Downloads/DESARROLLO_DE_APLICACIONES_EN_PYTHON_PAR.pdf

file:///C:/Users/Admin/Desktop/Numerical_Pyromaniacs_The_Use_of_Python_in_Fire_Re.pdf