

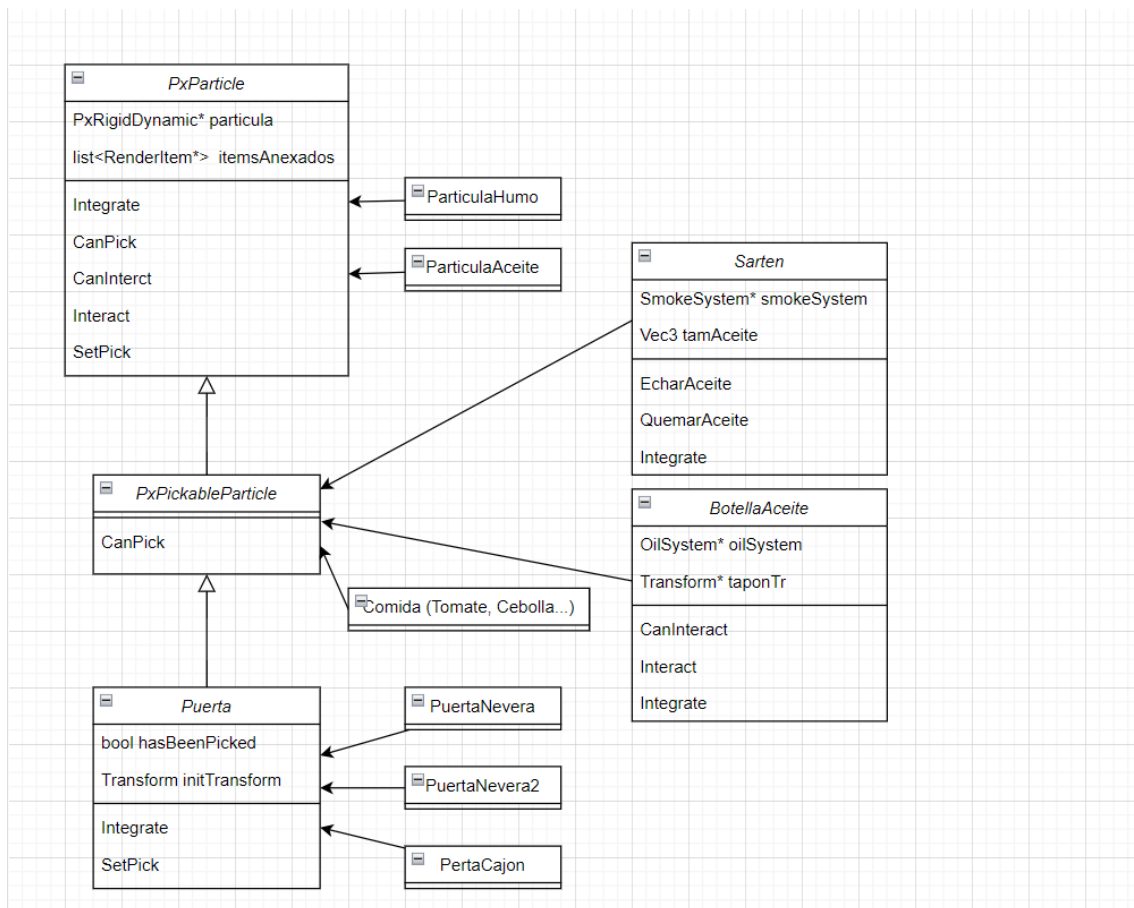
Memoria

El juego simula una cocina, en la que el jugador puede interactuar con los distintos objetos; cogiéndolos, dejándolos o tirándolos con los clics del ratón (más detalles del cómo hacerlo en el **Manual de Usuario**).

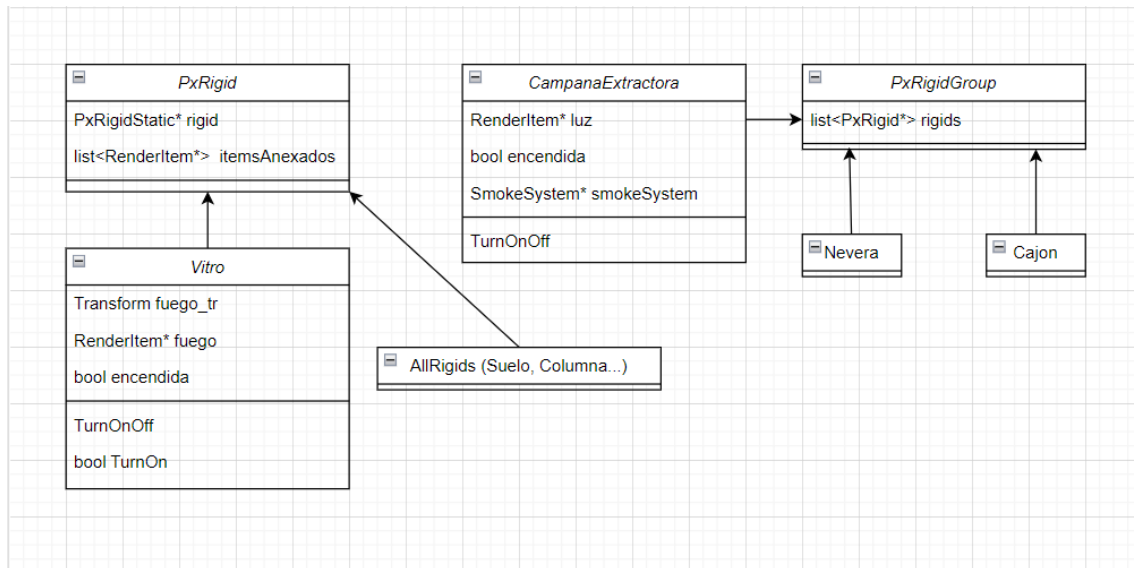
Todos los scripts nuevos, creados expresamente para realizar la práctica se encuentran en las carpetas que empiezan por *Px* (*PxForces*, *PxParticles* y *PxParticleSystem*). La escena está montada en */Scenes/ProyectoFinal.cpp* (y *.h*). Adicionalmente, se ha creado *CollisionCallback.h* y modificado ligeramente las clases *RenderUtils* y *Camera*, para implementar las colisiones y el movimiento en primera persona respectivamente.

Los objetos se dividen en:

- **Dynamic** (mi clase *PxParticle*). Que a su vez se divide en:
 - **Not Pickable**
 - **Pickable**. Adicionalmente se puede interactuar con *BotellaAceite* una vez cogida.

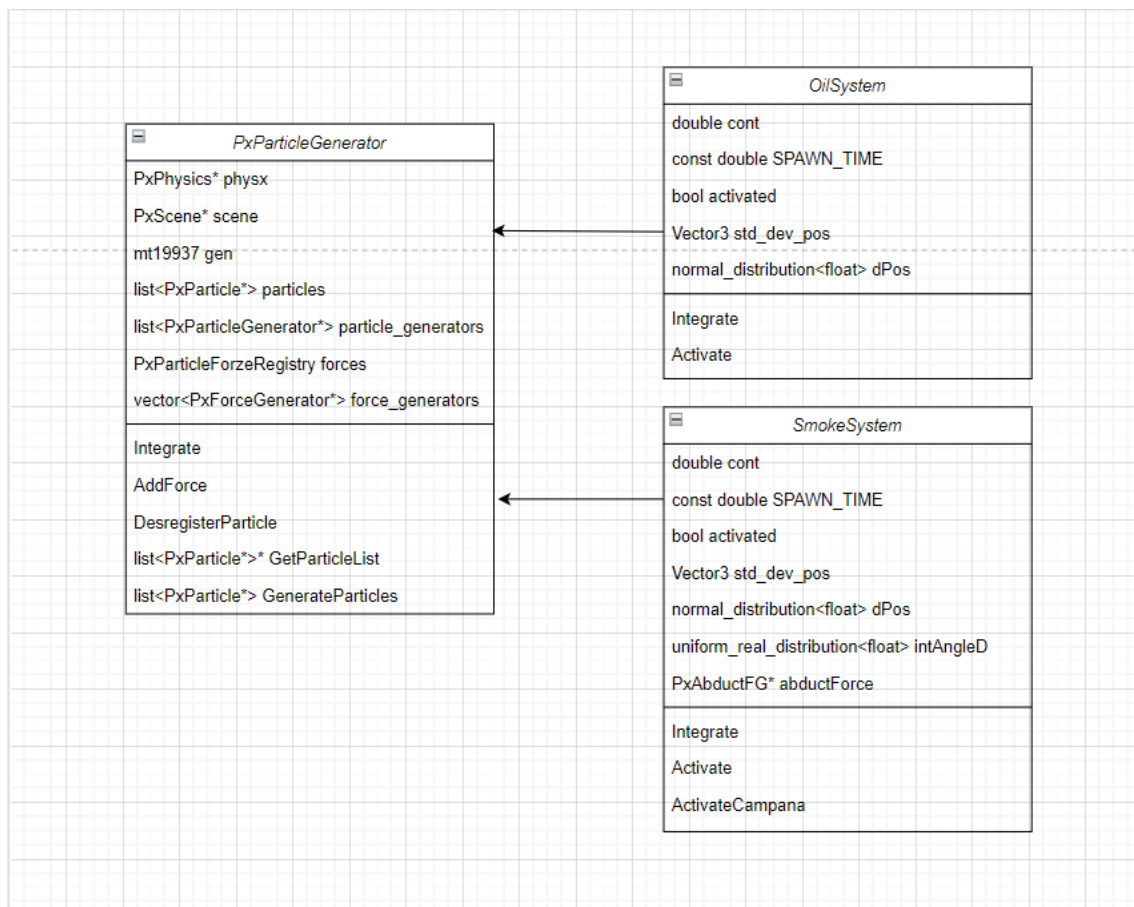


- **Static** (mi clase *PxRigid*). Que puede ser:
 - **Individual**
 - **Grupo de Estáticos** (*PxRigidGroup*)



He creado dos nuevos sistemas de sólidos:

- **Oil System:** se genera cuando interactúas (E) con la botella de aceite cuando la tienes cogida.
- **Smoke System:** se genera cuando la vitro está encendida (Z) y la sartén está encima de esta.



He creado tres fuerzas nuevas, las cuales se ven en acción el SmokeSystem:

- **PxFlowFG:** fuerza que es cada vez más tenue a medida que pasa el tiempo desde que una partícula de humo es creada.
 - $F.y = \text{humo} \rightarrow \text{flowF} * (1 - (\text{humo} \rightarrow \text{flowT} / \text{humo} \rightarrow \text{flowMaxT}))$
- **PxRotationFG:** las partículas de humo giran rápidamente en círculos alrededor de un eje imaginario con una función sin/cos.
 - $F = \text{Vector3}(\cos(\text{humo} \rightarrow \text{angle} * \text{PI} / 180.0f) * \text{humo} \rightarrow \text{radius}, \sin(\text{humo} \rightarrow \text{angle} * \text{PI} / 180.0f) * \text{humo} \rightarrow \text{radius}, 0)$
- **PxAbductFG:** las partículas son absorbidas hacia un punto fijo. Cuanto más cerca están del centro del área, con más fuerza son absorbidas. Esta fuerza se activa cuando la Campana Extractora está activada (X).
 - $F = (\text{humo} \rightarrow \text{pos} - \text{abductPos}).getNormalized() * (1 - (\text{dist} / \text{abductRange}) * \text{abductForce});$

Adicionalmente, he creado varios efectos visuales como la iluminación de las luces de la vitro o de la campana extractora cuando las activas (X y Z respectivamente). Además, si pruebas a echar aceite a la sartén este se quedará adherido, generando un “charco”.

Por último, una pequeña explicación de la implementación del sistema para coger rígidos y los objetos compuestos:

- Cuando coges un objeto este te seguirá a cierta distancia desde la **posición de la cámara** (Eye), teniendo en cuenta **dirección** en la que estás mirando (Dir).
- Se **corrige la rotación** del objeto que estás cogiendo, haciendo que este **mire** constantemente **hacia ti**.
- Los objetos compuestos están formados por un **PxRigidBody** y por un grupo de **RenderItems**, los cuales **copian** constantemente la **posición del primero** respecto a un **offset**.
- **Por defecto**, coges el objeto que se encuentre dentro del rango (una esfera imaginaria) que está **mas cerca de ti** cuando pulsas el clic **izquierdo**.
- Pero si pulsas la rueda del **medio del ratón**, se priorizarán los objetos que se encuentren **más cerca del centro de esa esfera** imaginaria. Puede resultar útil en algunas situaciones.
- Como ya se ha explicado en el Manual de Usuario, cuando tienes un objeto cogido, puedes **dejarlo caer** con clic **izquierdo** o **lanzarlo** con el clic **derecho**.