Los entornos de desarrollo jerárquicos se organizan en forma de árbol. De esta forma, siempre hay un nodo principal llamado raíz (*root*) y cada nodo tiene a su vez una serie de hijos (*children*). Las transformaciones geométricas que se realicen en el padre (*parent*) normalmente afectan a sus hijos.

En teoría explorarás en más detalle acerca de estos conceptos, pero por ahora bastarán para manejar la herramienta de forma básica.

1. Godot: Interfaz

Como se comentó al principio de esta guía, el software que usaremos para el desarrollo de nuestro entorno virtual será Godot. Godot trabaja sobre la base de un proyecto. Cada proyecto requiere un directorio vacío de comienzo.

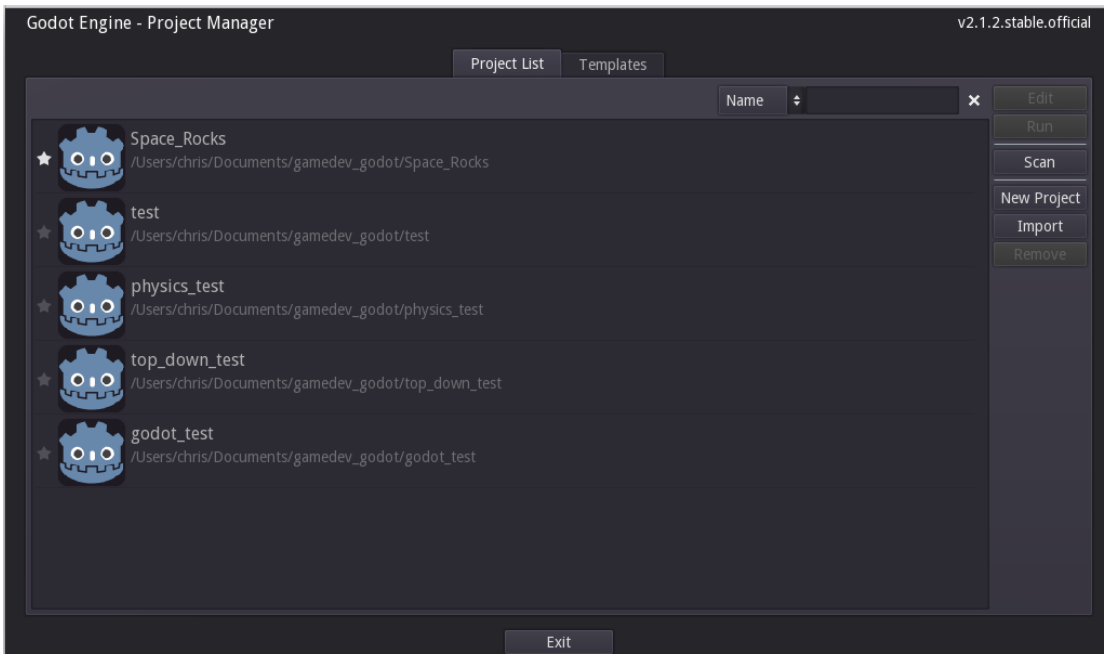
Un truco para no tener que importar el proyecto cada vez que arranques, es cargarlo directamente por línea de comandos:

Escribe desde una consola de tu sistema operativo:

```
./godot –help
```

para ver todas las opciones.

Si en algún momento se elimina un proyecto de la lista (pero no del disco) puede recuperarse simplemente importando dicha carpeta. Es por ello que es **FUNDAMENTAL** copiar la carpeta para hacer una **COPIA DE SEGURIDAD** de nuestro proyecto cada vez que terminemos la sesión de trabajo.



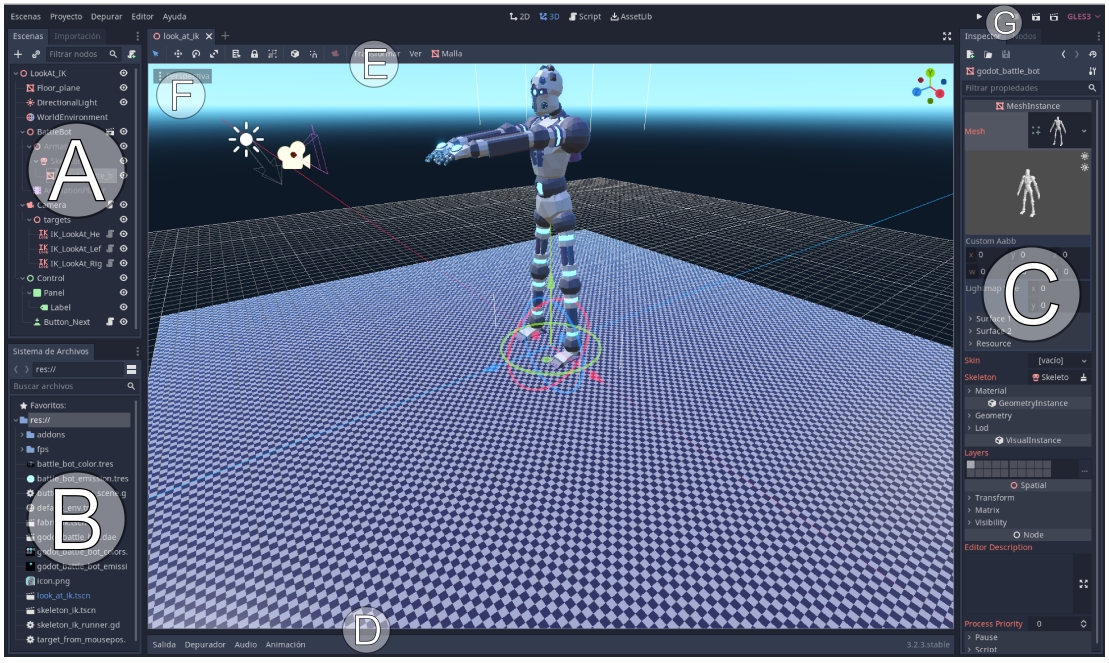
Por cierto, en este *enlace* tienes todas las teclas que por defecto están asignadas al editor de Godot.

Cada proyecto tendrá **escenas** y **recursos**.

- Las **Escenas** (*Scenes*): contendrán al resto de elementos y serán los que representen distintos mundos virtuales de nuestro proyecto.
    - Una *escena* puede formar parte de otra *escena*, como si fuera un nodo más.
    - También podemos transportarnos de una *escena* a otra mediante *portales* (*portals*). Es una forma muy cómoda de estructurar mundos virtuales muy complejos, pero ahora mismo no tienes que preocuparte por esto.
- Los **Recursos** (*Resources*): contendrán elementos importados en Godot, como pueden ser imágenes, audio, etc. También el conjunto de escenas que tengamos. Más tarde nos preocuparemos de ellos, al final de esta práctica, para gestionar escenas.

En cuanto a la interfaz principal, podemos ver un menú contextual (**C**) que cambiará según el nodo que hayamos seleccionado en el editor de nodos (**A**). Este interfaz puede ser complejo, y normalmente hay más elementos contextuales (**D**) que pueden desplegarse según las opciones que pulsemos en este menú.

Aquí tienes una breve descripción de todos los elementos que puedes ver en pantalla.



- **A.** La lista de nodos (*nodes*) es un menú contextual que indica los nodos que hay en la *Escena* abierta. Siempre habrá al menos un nodo raíz. Existen varios tipos de nodos (2D, 3D, interfaz, etc.).

- **B.** Lista de recursos (*resources*): es la lista de elementos que hay en el proyecto, solamente aparecerán los ficheros reconocidos por Godot, pero siempre podremos mostrar la carpeta con el botón derecho.
- **C.** El inspector es un menú contextual que indica las propiedades del elemento seleccionado.
- **D.** La ventana de herramienta especializada y depuración: esta ventana aparecerá con herramientas más avanzadas, como por ejemplo, con el editor de animaciones o las herramientas de depuración de código.
- **E.** La barra de herramientas (*Toolbar*) donde están todas las herramientas para mover, girar y escalar objetos, así como para cambiar las vistas del editor.
- **F.** La vista (*viewport*) es la representación de la escena en modo gráfico. Normalmente trabajaremos con escenas 3D, de tal forma que esta ventana contendrá, en la mayoría de los casos, la vista correspondiente a los elementos de una escena.
- **G.** Los botones de reproducción (*playtest buttons*) nos permiten probar una escena o todo el entorno.

Como todos los programas que están dirigidos también a artistas, la interfaz es muy configurable. ¡No te lies! fíjate bien en la forma que tiene cada uno de los apartados que hemos comentado. A menudo encontrarás información en la web que tienen invertidos los elementos (especialmente **A**, **B** y **C**) de la interfaz.

1.2 Tipos de nodos

Godot maneja cuatro tipos de nodo, que además vienen dados por colores:

- **Nodos 3D:** De color rosa.
- **Nodos 2D:** De color azul.
- **Nodos de interfaz de usuario:** De color verde.
- **Nodos especiales:** De color blanco.

Algo a considerar con los nodos es:

- Una escena siempre tiene un solo nodo raíz: puedes usar un nodo base simple (en una escena 2D sería un nodo *Node2D*, en una 3D sería un nodo *Spatial*) como nodo raíz.
- Las escenas pueden guardarse en disco y cargarse nuevamente más tarde. Cada vez que realices una modificación debes guardar la escena.
- Las escenas pueden ser instanciadas como nodos, y puede haber varias escenas en un proyecto. El anidamiento puede ser (en principio) tan profundo como queramos.
- Ejecutar un mundo virtual es ejecutar una escena. Por ello, lo primero que nos va a pedir el editor cuando le demos al botón de *Play* va a ser una escena principal para arrancar (más tarde podemos cambiarla).

- No deberíamos mezclar nodos (a menos que sepamos realmente que estamos haciendo) de distintos tipos.

1.3. Creación de nuestro primer proyecto, escena y nodos:

Ejercicio de entrenamiento 1:

> Lee y realiza el siguiente tutorial para crear tu primer proyecto de prueba.

> Puedes hacer click en este enlace para APRENDER más acerca de *los primeros pasos en Godot*.

Ejercicio de entrenamiento 2:

> Realiza el siguiente tutorial para crear tu primera escena **vacía** en 3D sobre el proyecto del Ejercicio 1.
> Ahora que ya sabes la base intenta crear una etiqueta y ejecutar *tu primer proyecto*.

2. Creación de entornos 3D, aprendiendo a usar Godot.

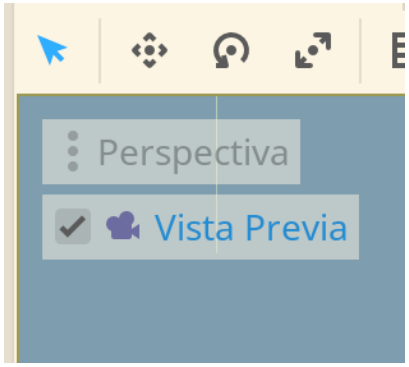
Primero vamos a poner cómoda la interfaz.

Una vez crees tu primera escena en 3D, ve al menú **Editor** → **Configuración del proyecto**, explora las opciones **Editor** y **Tema**, verás que puedes cambiar el idioma, el tamaño de la interfaz y los colores de la misma.

Es posible que tengas que reiniciar la aplicación tras algunos de los cambios.

Una vez hecho esto, necesitamos añadir un nodo de tipo Camera, y debes designarla como cámara principal. En toda escena solamente puede haber una cámara principal, pero en realidad podemos tener muchas cámaras. Busca la opción de **Cámara principal** (*current*).

Puedes girarla y manejarla con la interfaz (revisa el apartado **E** de la interfaz). También puedes visualizar una **vista previa** de la cámara **que tengas seleccionada** (revisa el apartado **F** de la interfaz).

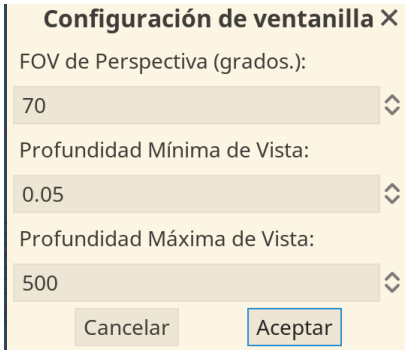


Puedes seguir este *tutorial* para ayudarte.

Ejercicio de entrenamiento 3:

> Plantéate cuál sería la utilidad de tener varias cámaras, pon ejemplos. Después, plantéale la duda a tu profesor de prácticas, ¡seguramente te sorprendas!

Todas las cámaras tienen un plano delantero y un plano trasero que recorta la visión (verás mejor esta parte en teoría), pero es posible que no veas los objetos lejanos debido a que el plano trasero está demasiado cerca. Para cambiarlo puedes ir al menú **Ver** → **Configuración...**, y después ajustar el último parámetro (**Profundidad máxima de vista**).

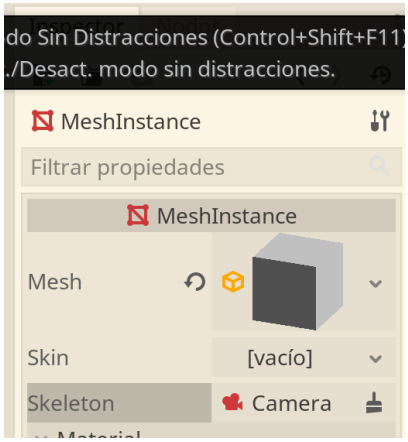


Ahora necesitamos una luz. Hay muchos tipos de luces en Godot (y en general en todos los entornos virtuales), pero por ahora solamente nos vamos a centrar en las luces de tipo **Omni**, basta saber que estas luces son un punto (invisible) en el espacio que emite luz en todas direcciones.

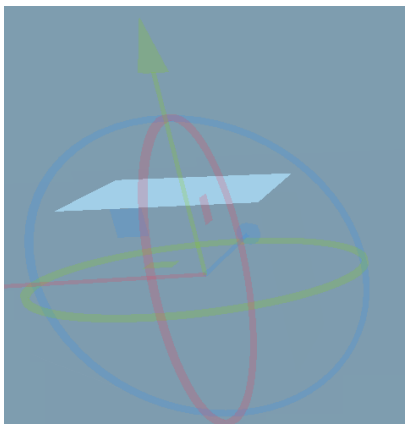
Busca el nodo **OmniLight** y añádelo en la escena. Ahora mismo puedes jugar con dos parámetros: **Color** y **Energía**, que cambian el color y la intensidad de la luz respectivamente.

RECUERDA: toda escena principal tiene que tener una cámara principal y al menos una fuente de luz.

Ahora ya podemos incluir nuestro primer modelo 3D. Godot tiene muchos elementos 3D, así que vamos a empezar con el modelo más sencillo (**MeshInstance**). Verás que el nodo añade algo que parece estar vacío. Eso es porque todo objeto 3D tiene que contenerse con geometría (vértices, aristas y caras), verás más de este tema en teoría, pero por ahora baste decir que podemos añadir geometría simple al nodo. En las opciones cambia la opción **Mesh** por un cubo.



Ahora deberías ver un cubo en la pantalla.



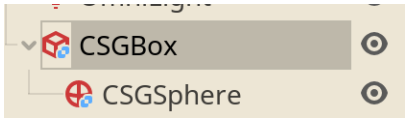
Ejercicio de entrenamiento 4:

> Crea un escenario de una ciudad sencillo (simulando que los cubos son rascacielos). Para ello juega con el escalado de los cubos y su posición.

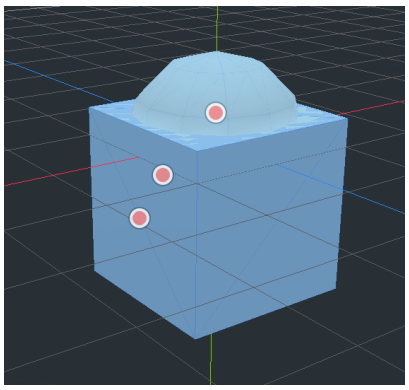
Ahora que ya sabes introducir elementos sencillos en 3D vamos a crear una escena algo más *real*. Para modelar objetos de forma sencilla podemos hacer uso de las operaciones booleanas. Verás más en detalle este tipo de operaciones en teoría, pero por ahora te puedes hacer una idea echando un vistazo a esta web: https://inkscape-manuals.readthedocs.io/en/latest/boolean-operations.html.

Estas operaciones booleanas se suelen utilizar en operaciones de sólidos, tipo CSG (verás más en teoría), donde se crea un arbol de operaciones. Para ello, se suele crear un nodo de operación y como hijos tenemos dos objetos que componemos. Las operaciones CSG normalmente incluyen una operación de unión, diferencia o intersección.

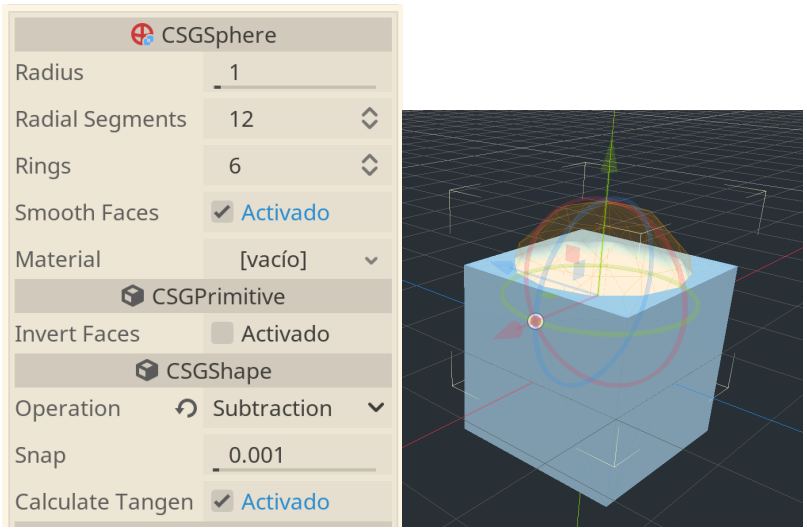
Crea un nodo de tipo **CSGCube** verás que en las opciones ya deja elegir el tipo de operación, ¡cuidado!, esta es la operación que realizará **sobre el objeto padre en su jerarquía**, deja aquí la **unión**. Ahora dentro de este nodo vamos a crear otro objeto, esta vez una **CSGSphere**, fíjate en la jerarquía y en que la operación de la esfera con respecto al padre también es una **unión**.



Ahora mueve la esfera para que se solape solamente parcialmente, verás como ambos objetos se funden en un único objeto.



Ahora cambia la operación de la esfera a **resta**, verás como automáticamente la esfera es restada del cubo.



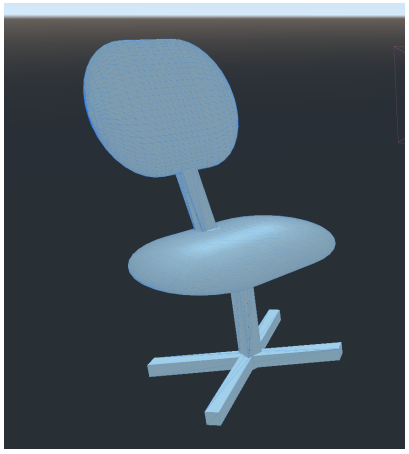
Ahora cambia la operación de **unión** por una **diferencia**.

3. Ejercicios de la primera parte de la práctica 1

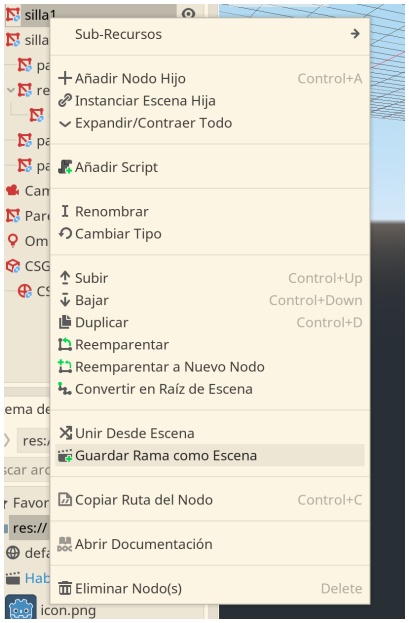
Realiza los siguientes ejercicios:

Ejercicio calificable P1. A.

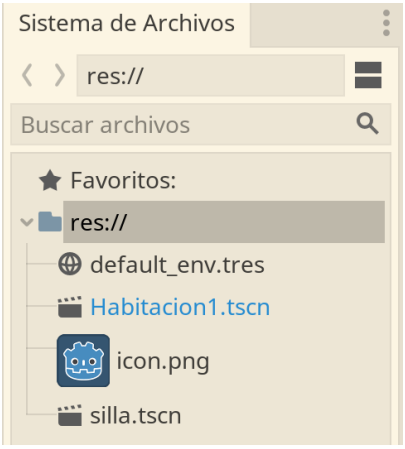
> Crea una silla utilizando todos los objetos de tipo CSG que quieras. Fíjate en la siguiente imagen para acercarte lo máximo posible a ella.



Fíjate que al pulsar sobre el botón derecho en un nodo, aparece un extenso menú, y una de las opciones permite convertir un nodo en una nueva escena.



Esta es una forma sencilla de exportar nodos como escenas, sin tener que complicarnos excesivamente para migrar trabajo ya realizado en la escena equivocada. Si lo haces, podrás comprobar como ahora tienes una nueva escena en los recursos:



Podemos añadir una instancia de una escena (también llamada *referencia*), como si fuera un nodo más, simplemente arrastrándolo y soltándolo en el árbol de nodos. Cualquier edición que hagamos en la escena referenciada, se aplicará a todas sus instancias de forma automática.

Ejercicio calificable P1. B.

> Crea una habitación mediante CSG y añade múltiples sillas dentro como instancias.

Ejercicio calificable P1. C.

> Ahora que sabes crear elementos sencillos, crea un bosquejo (dibujado en papel o en digital) de dos escenas distintas y un *pasillo* que las conecte pero que las habitaciones no se vean entre sí. Piensa un escenario que quieras modelar, y quédate con él para todas las prácticas.

> En realidad cada escena y el pasillo pueden ser cualquier cosa. Por ejemplo puedes hacer dos habitaciones y el *pasillo* de conexión puede ser una escalera en L. O puedes tener un bosque y una cueva, y el pasillo ser la entrada a la cueva. O un submarino y el puerto, y el pasillo ser la escalera ascendente y la escotilla de salida.

> Modela todos los elementos necesarios de la escena usando CSG. Se valorará especialmente la organización y la jerarquía de las escenas, y de sus objetos. Haz un buen uso de las instancias.

4. Iniciación a GDScript

Godot puede ser programado en múltiples lenguajes, pero en estas prácticas usaremos GDScript. Este lenguaje está específicamente diseñado para optimizar el diseño de escenas virtuales y la realización de *scripts* para juegos.

Ejercicio de entrenamiento 5:

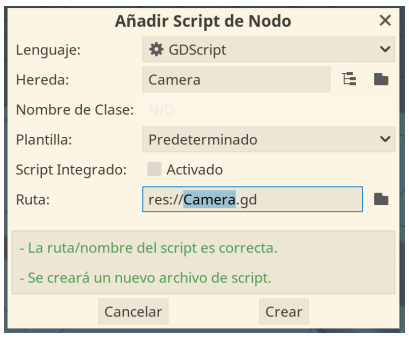
> Sigue el siguiente *tutorial* que compara GDScript con otros lenguajes que conoces.

> Puedes utilizar esta *referencia al lenguaje GDScript* como apoyo para entender el lenguaje.

4.1. Programación en Godot.

Godot permite añadir código a cualquier nodo que queramos. Vamos a tomar la cámara como ejemplo. Selecciona la cámara, y en el menú que aparece al pulsar el botón derecho del ratón, selecciona **Añadir Script**.

Aparecerá entonces un menú como el siguiente:



en el que insta a crear un nuevo Script, heredando del objeto **Camera**, e indicando que el *script* se grabará como recurso. Esto es porque todos los *scripts* son recursos asociados a nuestro proyecto, y aunque se pueden integrar con los nodos, conviene no hacerlo para poder editarlos de forma independiente.

Tras hacer estos pasos, aparecerá el código asociado a la cámara (fíjate en el icono del pergamino al lado del nodo).

```
1  extends Camera
2
3
4  # Declare member variables here. Examples:
5  # var a = 2
6  # var b = "text"
7
8
9  # Called when the node enters the scene tree for the first time.
10 func _ready():
11     pass # Replace with function body.
12
13
14 # Called every frame. 'delta' is the elapsed time since the previous frame.
15 #func _process(delta):
16 #    pass
17
```

Primero, indicar que la función (o mejor dicho, método) _ready() está presente en todos los objetos, y que se actualiza al iniciarse el objeto la primera vez. Mientras que la función (o mejor dicho, método) _process(), se llama cada fotograma, y toma un parámetro que es el número de segundos desde la última llamada.

Ahora no podemos hacer demasiado sin conocer la funcionalidad de la cámara. Vamos a pulsar en el botón derecho sobre el nodo de la cámara y pulsar **Abrir Documentación**. Verás que aparece información muy útil sobre el nodo, así como todos los métodos de la clase, de forma que podamos programarla:

Clase: Camera

Hereda: Spatial < Node < Object

Heredada por: ARVRCamera , ClippedCamera , InterpolatedCamera

Camera node, displays from a point of view.

Descripción

Camera is a special node that displays what is visible from its current location. Cameras register themselves in the nearest Viewport node (when ascending the tree). Only one camera can be active per viewport. If no viewport is available ascending the tree, the camera will register in the global viewport. In other words, a camera just provides 3D display capabilities to a Viewport, and, without one, a scene registered in that Viewport (or higher viewports) can't be displayed.

Propiedades

KeepAspect	keep_aspect [predeterminado: 1]
int	cull_mask [predeterminado: 0xFFFFF]
Environment	environment
float	h_offset [predeterminado: 0.0]
float	v_offset [predeterminado: 0.0]
DopplerTracking	doppler_tracking [predeterminado: 0]
Projection	projection [predeterminado: 0]
bool	current [predeterminado: false]
float	fov [predeterminado: 70.0]

Además en la parte superior tenemos el esquema de la herencia completa, así tenemos que:

- **Cámara ← Spatial ← Node ← Object**

Ejercicio de entrenamiento 6:

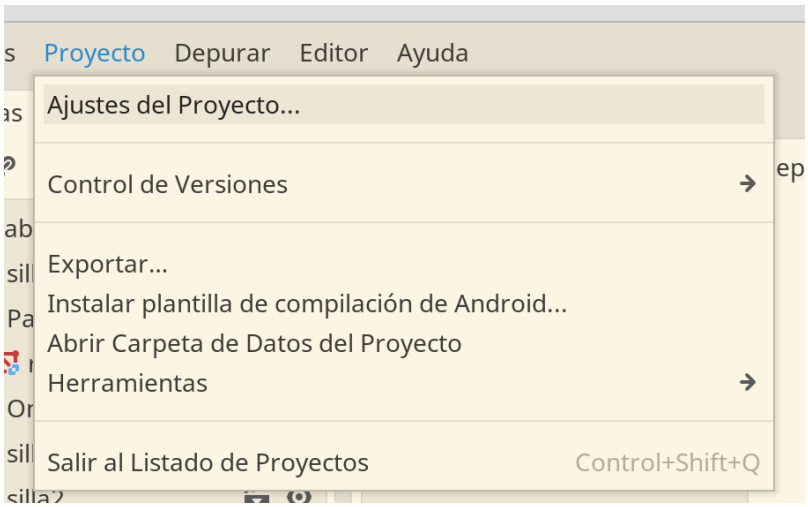
> Busca entre los ancestros de la cámara un método para rotar, y rota cada vez que se llame a la función _process 0.01 grado en el eje *y*. Recuerda que puedes crear variables globales en la parte superior del código (revisa los comentarios).

> Ahora haz los cambios necesarios para que rote un grado cada décima de segundo que pase. Plantea cómo podrías resolver el problema de que vaya *a saltos*.

4.2. Definición y uso de teclas

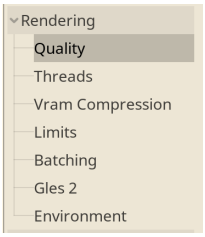
Godot permite el uso de dispositivos de interacción de forma abstracta, de tal forma que no tendremos que preocuparnos de realizar a mano el mapeo con los dispositivos de interacción.

Pero antes, vamos a ver como configurar el proyecto. Puedes ir al menú **Proyecto** → **Ajustes del proyecto...**.



Esto desplegará todas las opciones del proyecto, incluyendo la escena que será considerada como principal, el icono de la aplicación, etc.

Las opciones más relevantes en cuanto a gráficos y resolución se encuentran en los menús de **Rendering** y **Display**.

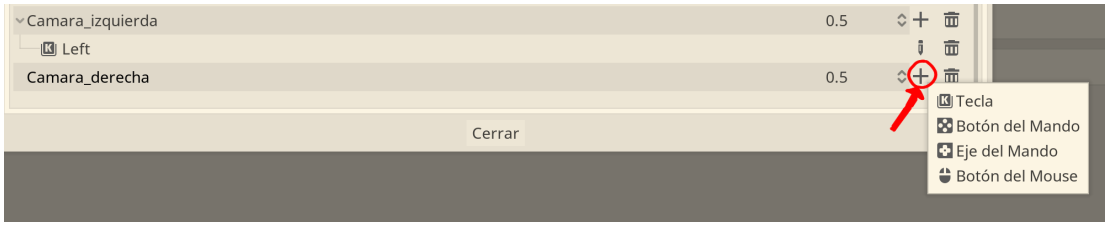


Ejercicio de entrenamiento 7:

> Deja un rato el cursor sobre las opciones, verás una ayuda acerca de las mismas. Revisa las opciones de **Display** → **Window** y pregunta a tu profesor lo que no entiendas de estas opciones.

Volviendo a los **Ajustes del proyecto...**, verás que en la parte superior hay una pestaña llamada **Mapa de entrada**, si pulsas ahí verás una serie de teclas definidas y acciones.

Prueba a crear un par de acciones, por ejemplo «Cámara izquierda» y «Cámara derecha» y asígnale los cursores (←, →).



Godot permite manejar eventos en base a acciones de forma muy simple. Echa un vistazo a la *clase Input* antes de intentar el siguiente ejercicio.

Ejercicio calificable P1. D.

> Crea una cámara orbital en tu proyecto:

> Con ayuda de la jerarquía y de los nodos **Point3D**, los cuales permiten transformaciones geométricas (pero son invisibles), añade dos acciones para girar y rotar la cámara en dos ejes distintos. Piensa bien la jerarquía a montar antes de realizar ningún *script*, ayúdate del documento de matemáticas sobre vectores y de las transparencias de teoría de matrices homogéneas si no lo tienes claro. Asigna los cuatro cursores a las acciones de giro de cámara.

> Limita los ángulos máximos y mínimos para evitar que la cámara se invierta.