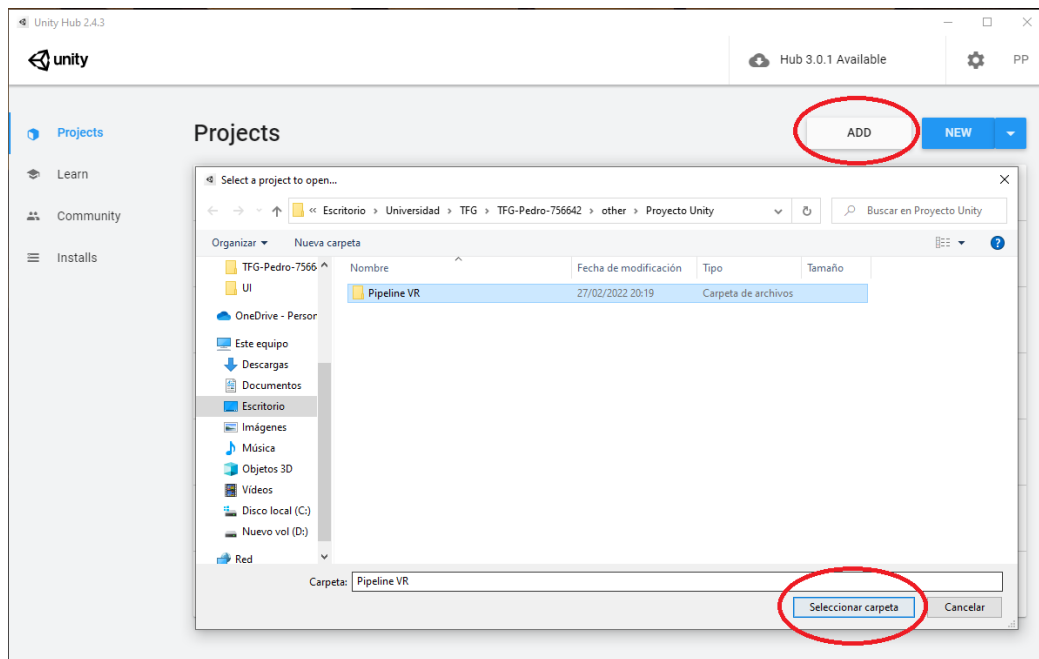


Requisitos previos para poder ejecutar correctamente el pipeline

- Disponer de unas gafas VR HTC Vive Pro con el add-on para *eyetracking* de PupilLabs.
- Tener instalado Unity Hub (cualquier versión) y el editor de Unity en su versión 2020.3.25f1, se suelen instalar conjuntamente al utilizar el instalador proporcionado por Unity.
Se puede obtener en <https://unity3d.com/es/get-unity/download/archive>.
Si se tiene una versión más reciente del editor de Unity podría funcionar igualmente, pero Unity necesitará un tiempo para actualizar el proyecto a la versión del editor.
- Visual Studio 2019 en adelante (muy recomendable pero no obligatorio, además el instalador de Unity permite instalarlo desde ahí directamente)
- Tener instalado SteamVR en el equipo.
- Tener instalado Steamworks common redistributables (se instalan desde el propio programa de Steam automáticamente al descargar cualquier programa que los necesite, como por ejemplo SteamVR)
- Tener instalada la aplicación de *eyetracking* de PupilLabs. Se adjunta el instalador en la carpeta /other/Instalador Aplicación PupilLabs/ y si se desea, se puede obtener en <https://github.com/pupil-labs/pupil/releases/tag/v3.5>

Instalación y puesta a punto del pipeline en Unity

1. El proyecto completo se encuentra en la carpeta /other/Proyecto Unity/. Se trata de una carpeta llamada "Pipeline VR" que contiene todos los archivos del proyecto en el momento de su entrega final. Se proporciona de esta forma y no como paquete de Unity porque así vienen incluidos los Plugins y otros paquetes que han sido necesarios para este proyecto, eliminando la necesidad de instalarlos aparte.
2. Se debe copiar la carpeta "Pipeline VR" junto con todo su contenido a un directorio cualquiera del equipo, allí donde se quiera tener almacenado el proyecto.
3. Abrir Unity Hub, el lanzador de Unity que se instala junto con el editor del motor. Allí, en el menú de proyectos se selecciona la opción "Add" (Unity Hub 2.X, en 3.0 hay que ir a la flecha junto al botón "Open" y seleccionar "Add Project from disk") y se abrirá una ventana para elegir una carpeta del sistema.
4. En esa ventana, ir a la ubicación de la carpeta "Pipeline VR", seleccionar la propia carpeta y hacer clic en "Seleccionar carpeta". Ahora se listará en la lista de proyectos de Unity Hub.



5. Hacer clic en el proyecto en la lista. Esto abrirá el editor de Unity. Para minimizar el peso de los archivos incluidos aquí, se han eliminado algunas carpetas con binarios que Unity necesita, por lo que la primera vez que se abra el proyecto puede tardar varios minutos en completar la operación, debido a que estará regenerando estos archivos sin necesidad de interacción durante el proceso.
6. Si todo ha salido bien se debería haber clonado satisfactoriamente el proyecto.

Utilización del pipeline

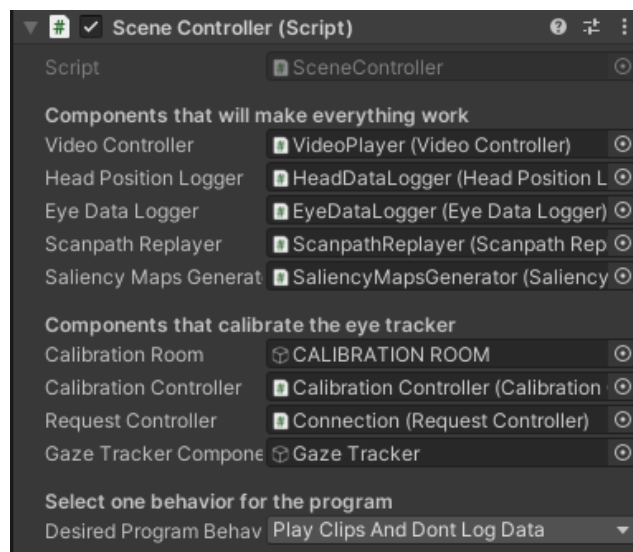
El pipeline se puede utilizar para diferentes propósitos, por lo que se comienza viendo la forma de configurar el componente que nos permite elegir entre estos diferentes modos de funcionamiento, y a continuación se explicará la forma de configurar el resto de componentes:

Configurar el componente Mode Selector

Se debe seleccionar el objeto llamado "MODE SELECTOR" en la jerarquía de la escena. A continuación se mostrarán sus componentes en el inspector de Unity. Nos interesa el componente "Scene Controller". Se trata de un script que gestiona el comportamiento del pipeline, por lo que se debe elegir aquí el que se desea.

Las propiedades del componente vistas en el inspector de Unity son:

- VideoController: Debe referenciar al componente de este tipo en la escena.
- Head Position Logger: Debe referenciar al componente de este tipo en la escena.
- Eye Data Logger: Debe referenciar al componente de este tipo en la escena.
- Scanpath Replayer: Debe referenciar al componente de este tipo en la escena.
- Saliency Maps Generator: Debe referenciar al componente de este tipo en la escena.
- Calibration Room: Debe referenciar a un objeto que como hijos tenga los objetos que representan la “habitación” donde se calibra el *eyetracker*. Se necesitan cuatro paredes, techo y suelo que oculten la *skybox*, debido a que esta opacaría por completo la interfaz del proceso de calibración si no la taparan las paredes de esta “habitación”. Para detalles de este funcionamiento, consultar el vídeo [/other/Defensa/Videos/Funcionamiento del experimento desde el editor.mp4](#)
- Calibratation Controller: Debe referenciar al componente de este tipo en la escena.
- Request Controller: Debe referenciar al componente de este tipo en la escena.
- Gaze Tracker Component: Debe referenciar al objeto de la escena llamado “Gaze Tracker”, contiene diversos componentes relativos al plugin de Pupillabs, se recomienda no modificarlo.
- Desired Program Behavior: Comportamiento deseado para el pipeline cuando se ejecute.



Hay varios modos de funcionamiento disponibles para el pipeline:

- Reproducir vídeos y no crear logs
- Reproducir vídeos y crear logs
- Modo repetición (recreación de un scanpath)
- Generar mapas de saliencia

NOTA: Aunque se configure el programa para trabajar en un modo (por ejemplo, calcular mapas de saliencia) puede suceder que el no haber configurado opciones relativas a otros

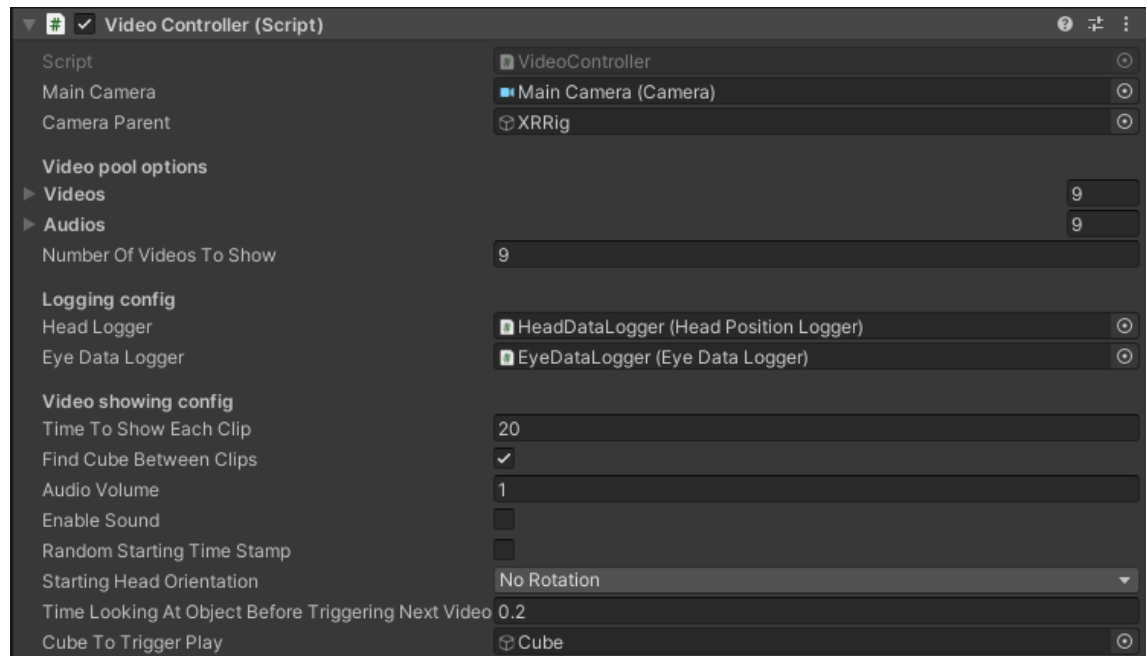
modos de funcionamiento (por ejemplo, dejar como nulo el log a cargar para el modo repetición) desencadene errores, para evitarlos se recomienda dar valores válidos a todos los parámetros de los componentes de logging, cálculo de mapas de saliencia y recreación de scanpaths; aunque no se necesiten algunos componentes para una ejecución.

Configurar el componente VideoController

Este componente se encarga de la reproducción de los clips de vídeo y de audio durante el proceso de recopilación de datos. Las opciones que el componente permite configurar respecto a esta reproducción son las siguientes:

- Main Camera: Debe referenciar a la cámara de la escena; solo debería haber una, dentro de la jerarquía del objeto XRRig.
- Camera Parent: Debe referenciar al objeto raíz de la jerarquía de la cámara. Esto se utiliza para forzar una rotación en el punto de vista de los observadores.
- Videos: Lista conteniendo los clips de vídeo que se reproducirán seleccionados de manera aleatoria.
- Audios: Lista conteniendo los clips de audio que se reproducirán. **Nota:** Los elementos de ambas listas deben corresponder en orden, es decir, el audio en primer lugar corresponderá con el vídeo en primer lugar.
- Number of videos to show: Número de vídeos que reproducir de entre los seleccionados. Se pueden tener 10 vídeos pero solo reproducir 4, que son elegidos de manera aleatoria.
- Head Logger: Debe referenciar al componente HeadLogger.
- Eye Data Logger: Debe referenciar al componente EyeDataLogger.
- Time to show each video: Segundos durante los cuales reproducir cada vídeo antes de pasar al siguiente. (Se recomienda en torno a 20 segundos o mayor)
- Find cube between clips: Determina la forma de pasar de reproducir un vídeo a otro. Si está activado, los usuarios deben buscar un objeto a su alrededor con la mirada para que se comience a reproducir el siguiente vídeo. Si no, se esperará un tiempo fijo antes de reproducir el siguiente vídeo.
- Audio Volume: Valor entre 0 y 1 para ajustar el volumen del sonido.
- Enable Sound: Activar o desactivar el sonido. Esta opción sobreescrive la anterior.
- Random starting timestamp: Si está activado, se comenzará a reproducir cada vídeo en un instante aleatorio, dejando margen para que se puedan reproducir tantos segundos de vídeo como se han indicado. Si no, se reproducirán desde el principio.
- Starting head orientation: Determina si se reubica el campo visual de los observadores al inicio de cada vídeo. Si se elige “No Rotation”, no se reorienta al usuario. Elegir “Randomize 90 Deg Intervals” hará que el campo visual del usuario esté centrado a 0, 90, 180 o 270 grados en longitud, mientras que “Fully Randomized” genera una orientación inicial aleatoria tanto para latitud como para longitud al comienzo de cada vídeo.
- Time looking at object before triggering next video: Cantidad de tiempo que se debe observar el objeto posicionado aleatoriamente alrededor de los observadores antes de reproducir el siguiente vídeo.

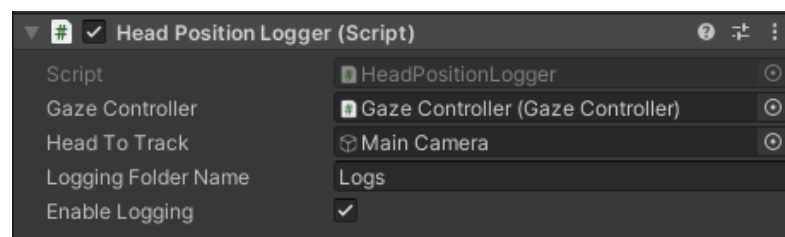
- Cube to trigger play: Referencia a un objeto, con la forma que tendrá el objeto que deben buscar los observadores entre escenas, en caso de haber activado “Find cube between clips”.
- Time between clips: Si “Find cube between clips” está desactivada, es la cantidad fija de tiempo a esperar antes de iniciar la reproducción del siguiente vídeo.



Configurar el componente HeadDataLogger

Este componente permite recolectar información de la orientación de la cabeza de los usuarios y la vuelca a ficheros .xml. Los detalles acerca del proceso y el formato se pueden encontrar en la memoria. Las opciones del componente son:

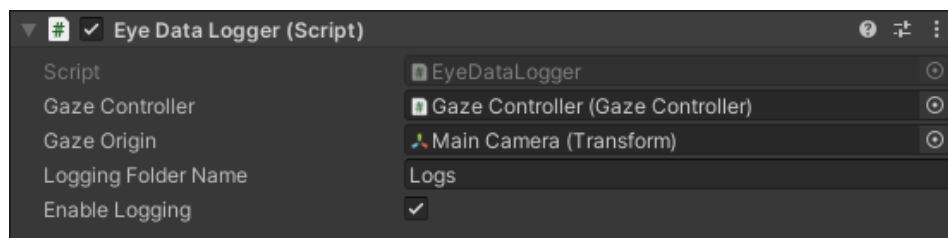
- Gaze Controller: Debe referenciar al componente de igual nombre, en el objeto de igual nombre, hijo de “Gaze Tracker” en la jerarquía.
- Head to track: Debe referenciar a la cámara principal (y única) de la escena.
- Logging folder name: Nombre de la carpeta a la que volcar los logs. Por ejemplo “Logs/Pedro” o “Logs/Sandra”.
- Enable logging: Habilita o deshabilita el proceso de logging. Debe dejarse activo, serán otros componentes los que cambien su valor cuando sea necesario.



Configurar el componente EyeDataLogger

Este componente permite recolectar información de la orientación de la mirada de los usuarios y la vuelca a archivos .xml. Los detalles acerca del proceso y el formato se pueden encontrar en la memoria del Trabajo. Las opciones del componente son:

- Gaze Controller: Debe referenciar al componente de igual nombre, en el objeto de igual nombre, hijo de “Gaze Tracker” en la jerarquía.
- Gaze Origin: Debe referenciar al componente Transform de la cámara principal (y única) de la escena.
- Logging folder name: Nombre de la carpeta a la que volcar los logs. Por ejemplo “Logs/Dani” o “Logs/Ana”.
- Enable logging: Habilita o deshabilita el proceso de logging. Debe dejarse activo, serán otros componentes los que cambien su valor cuando sea necesario.

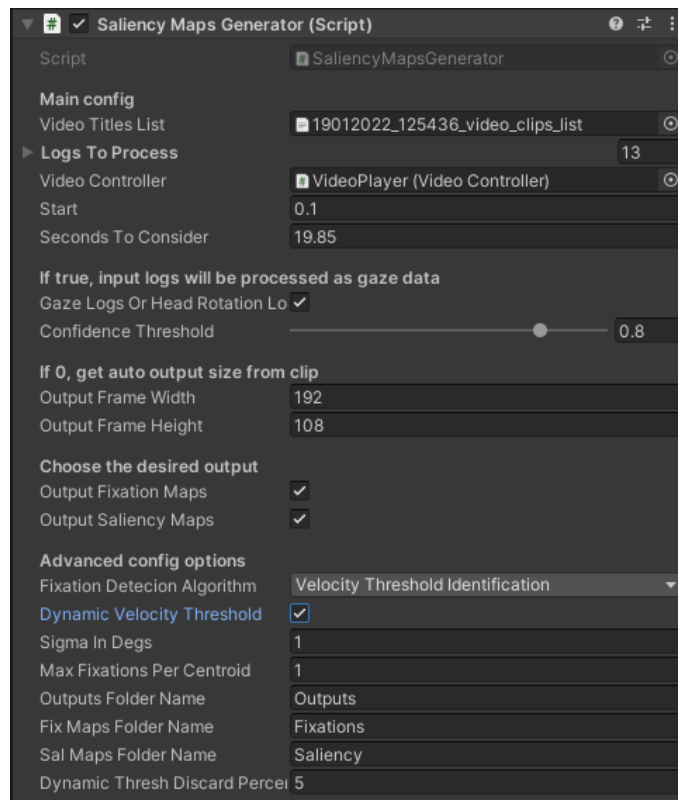


Configurar el componente SaliencyMapsGenerator

Este componente se encarga de generar los mapas de saliencia y de fijaciones. Sus opciones de configuración abarcan los logs a utilizar para el proceso, el periodo temporal a considerar, o los parámetros del algoritmo de generación de los mapas. Estas opciones son:

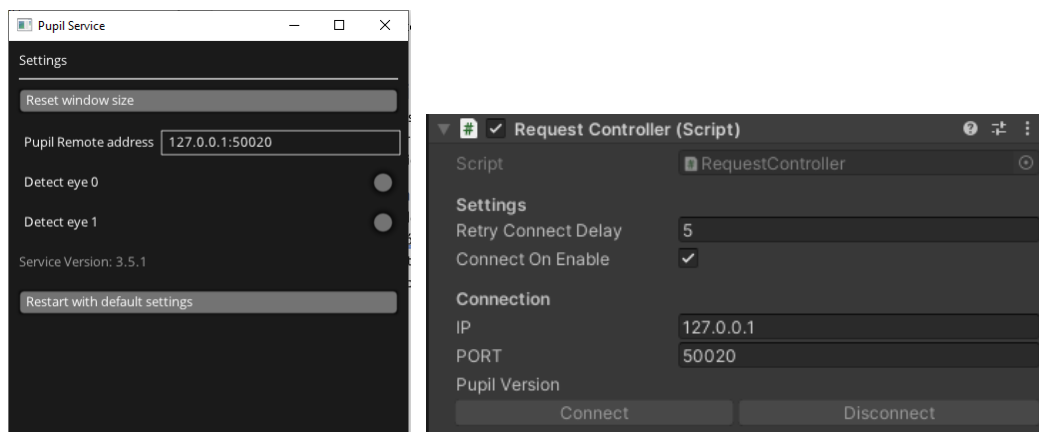
- Video titles list: Fichero de texto con la lista de títulos de los clips de vídeo. Se recomienda cargar el .txt correspondiente al primero de los logs que se quieran procesar.
- Logs to process: Lista de logs a procesar. Se admiten ficheros con datos de la cabeza y de la mirada, de manera exclusiva entre ambos tipos. Se pueden cargar varios logs correspondientes a diferentes observadores para calcular las zonas salientes en promedio. Además, se **debe** garantizar que todos los logs cargados corresponden al mismo vídeo. Para ello, se puede consultar el número de clip en los nombres de los ficheros, junto con sus respectivos .txt que contienen la lista ordenada de títulos.
- VideoController: Debe referenciar al componente VideoController.
- Start: Instante del tiempo a partir del cual calcular mapas de saliencia, medido en las muestras registradas en los logs.
- Seconds to consider: Número de segundos que tener en cuenta para el cálculo de los mapas.
- Gaze logs or head rotation logs: Indica si se están cargando logs de datos de la cabeza o de la mirada. Afecta al procesamiento interno de los datos.

- Confidence threshold: Umbral de confianza para tener o no en cuenta las medidas del *eyetracker* si los datos son de la mirada.
- Output frame width: Anchura de los mapas de saliencia generados. Si se le da valor 0, se utilizan las dimensiones de los frames del vídeo.
- Output frame height: Altura de los mapas de saliencia generados. Si se le da valor 0, se utilizan las dimensiones de los frames del vídeo.
- Output fixation maps: Generar o no generar mapas de fijaciones.
- Output saliency maps: Generar o no generar mapas de saliencia.
- Fixation detection algorithm: Algoritmo de detección de fijaciones a utilizar. Por el momento solo IVT.
- Dynamic velocity Threshold: Habilitar o no el cálculo del umbral de velocidad de forma dinámica.
- Sigma in degs: Sigma de la convolución Gaussiana, en grados.
- Max fixations per centroid: Número máximo de fijaciones que colapsar en una fijación centroide, que se usará para generar los mapas.
- Outputs folder name: Nombre de la carpeta donde se colocarán los ficheros generados. Allí se generará una carpeta con el nombre del vídeo que puede contener dos subcarpetas: una para cada tipo de mapas.
- Fix maps folder name: Nombre de la carpeta de los mapas de fijaciones.
- Sal maps folder name: Nombre de la carpeta de los mapas de saliencia.
- Dynamic threshold discard percent: Porcentaje de velocidades angulares más altas que descartar. Parámetro del cálculo del umbral para IVT de forma dinámica. Solo es visible en el editor si se tiene activada la opción “Dynamic velocity Threshold”.
- Velocity Threshold IVT: Umbral fijo de velocidad angular para IVT, en grados por segundo. Solo es visible en el editor si se tiene desactivada la opción “Dynamic velocity Threshold”.



Cómo iniciar una recogida de datos

1. Importar los vídeos y audios que se deseen utilizar para el experimento al proyecto de Unity. En la jerarquía del proyecto se tienen las carpetas “Assets/360Videos” y “Assets/AmbisonicAudio” para vídeos y audio ambisónico, pero se pueden utilizar otras si se desea.
2. Iniciar SteamVR (puede que no sea necesario y se inicie al ejecutar todo)
3. Conectar adecuadamente las gafas y el dispositivo de *eye tracking* al equipo.
4. Iniciar la aplicación Pupil Service.
5. Seleccionar en la jerarquía el objeto Connection, hijo del objeto Gaze Tracker. En el componente Request Controller, cambiar la dirección IP y el puerto para que coincidan con los de PupilService. También se pueden fijar aquí y cambiar en Pupil Service.



6. Seleccionar en la jerarquía el objeto VideoPlayer y configurar las opciones del componente VideoController en el inspector, ignorando el resto de componentes.
7. Seleccionar en la jerarquía el objeto HeadDataLogger y editar las opciones del componente HeadPositionLogger.
8. Seleccionar en la jerarquía el objeto EyeDataLogger y editar las opciones del componente de igual nombre, asociado al objeto.
9. Seleccionar en la jerarquía el objeto MODE SELECTOR y en el componente Scene Controller, editar el desplegable “Desired Program Behavior” haciendo que indique “Play Clips And Log Data”.
10. Comenzar la ejecución del programa y esperar a que termine (el comportamiento de un experimento varía en función de las opciones elegidas, se describe con mayor

precisión en la memoria del Trabajo)

11. Los ficheros de log se habrán generado en las carpetas especificadas al configurar los componentes de logging. Si se accede a ellas usando el navegador de ficheros del editor de Unity, es posible que no aparezcan, en ese caso se recomienda hacer Ctrl+r (refrescar). Puede tardar unos segundos en importar los ficheros de log al proyecto.
12. Adicionalmente se habrán generado dos ficheros .txt que contienen en cada línea el nombre de los vídeos y de los audios, respectivamente. Estarán en el mismo orden que se haya especificado en el editor de Unity.

NOTA: Un vídeo que ejemplifica el proceso de puesta a punto y recogida de datos para un usuario se puede encontrar en /other/Ejemplo de uso modo captura de datos.mp4

Cómo iniciar una recreación de Scanpaths

1. Importar al proyecto los ficheros de logging cuyos scanpaths se desee recrear. Para ello, basta con arrastrar hasta la ventana de ficheros del proyecto los ficheros que se deseen importar. Si ya están importados, se puede saltar al siguiente paso.
2. Seleccionar el objeto ScanpathReplayer en la jerarquía y configurar su componente de igual nombre.
3. Seleccionar en la jerarquía el objeto MODE SELECTOR y en el componente Scene Controller, editar el desplegable “Desired Program Behavior” haciendo que indique “Log Based Replay Mode”.
4. Ejecutar el pipeline y observar la recreación de los scanpaths especificados.

NOTA: Un vídeo que ejemplifica el proceso de puesta a punto y recreación de scanpaths se puede encontrar en /other/Ejemplo de uso modo replay.mp4

Cómo calcular mapas de fijaciones y de saliencia

1. Importar al proyecto los ficheros de log que se deseen procesar. Para ello, basta con arrastrar hasta la ventana de ficheros del proyecto los ficheros que se deseen importar. Si ya están importados, se puede saltar al siguiente paso.
2. Seleccionar el objeto SaliencyMapsGenerator en la jerarquía y configurar las opciones del componente de igual nombre asociado al objeto.
3. Seleccionar en la jerarquía el objeto MODE SELECTOR y en el componente Scene Controller, editar el desplegable “Desired Program Behavior” haciendo que indique “Compute Saliency Maps”.

4. Comenzar la ejecución del programa. No se mostrará nada en pantalla, pero en la consola de mensajes del editor de Unity se informa del progreso actual del proceso.
5. Cuando termine de ejecutar, se habrán generado los mapas de fijaciones y de saliencia en las carpetas especificadas. Si se accede a ellas usando el navegador de ficheros del editor de Unity, es posible que no aparezcan, en ese caso se recomienda hacer Ctrl+r (refrescar). Puede tardar unos segundos o incluso unos pocos minutos en importar los mapas de saliencia y fijaciones, dependiendo de la cantidad generada y de sus dimensiones.

NOTA: Un vídeo que ejemplifica el proceso de puesta a punto y generación de mapas de saliencia y fijaciones se encuentra en [/other/Ejemplo generación mapas de saliencia.mp4](#)