

Cuaderno Bitácora TFM

José Carlos Rodríguez Cortés

29 de junio de 2017

Resumen

Cuaderno en el que se apuntarán los eventos sucedidos durante cada día de investigación para el TFM para su posterior estudio y/o integración a la memoria final del trabajo, así como al artículo.

1. Día 1: 20/06/2017

1.1. Hechos

Toma de contacto con Oculus SDK y Unity. Implementación de los algoritmos preparados con anterioridad. A saber:

- Algoritmo de movimiento mediante raycast. Implementando para ello 3 tipos de movimientos:
 - Teletransporte
 - Desplazamiento bidimensional sobre un plano con tiempo constante, es decir, sea cual sea la distancia a recorrer siempre se tardará lo mismo, por lo que algunos movimientos serán más acelerados que otros.
 - Desplazamiento tridimensional. Se ejecuta un desplazamiento bidimensional con un "salto" implementado a partir de una función senoidal con una altura máxima. También tiene una duración constante por lo que puede ser un movimiento más veloz si apuntamos a zonas distantes del mapa.
- Algoritmo de foco dinámico sin necesidad de eyetracking utilizando para ello el artículo [Porcino et al.].

1.2. Observaciones

Tras hacer algunas pruebas de compatibilidad de estos algoritmos con las gafas Oculus Rift se prevee buen funcionamiento de todos ellos para escenarios más complejos. A falta de testear estos algoritmos con más personas se asumirá que funcionan y no causan fatiga.

Se ha detectado que en el enfoque dinámico el paso de enfocado a desenfocado y viceversa es demasiado brusco.

1.3. Trabajo Futuro

Para el próximo día se plantean las siguientes tareas:

- Implementar interacción de agarrar y soltar. Idea: Utilizar OVRGrabber y OVRGrabbable.
- Intentar implementar manos virtuales a partir de modelos gratuitos de internet para mejorar la experiencia dentro del mundo virtual.

2. Día 2: 22/06/2017

2.1. Hechos

Ajustados algunos parámetros de los algoritmos probados el Día 1. Sin cambios observables en la visualización final.

Se intenta implementar funcionalidad para agarrar objetos mediante el OVRGrabbable y OVRGrabber de la librería OVR de Oculus SDK, sin embargo, durante la búsqueda de información para esta implementación se encuentra un SDK que abstrae el oficial de Oculus llamado NewtonVR.

Se despliega una escena sencilla con este nuevo SDK y se hacen pruebas de interacción. Se integra Avatar SDK para potenciar la visualización de la escena y la inmersión del jugador.

Tras hacer algunas pruebas con el Avatar SDK en escenas independientes se integra Avatar SDK a la escena realizada con NewtonVR resultando en una escena en la que podemos ver nuestras manos virtuales además de interactuar con algunos objetos a los que se les aplican físicas.

2.2. Observaciones

Tras utilizar NewtonVR para implementar una escena sencilla se observa que es mucho más sencillo crear escenas con este nuevo SDK que con el SDK básico de Oculus por lo que se plantean nuevos desarrollos a partir del mismo.

Avatar SDK es extremadamente sencillo de incorporar a cualquier escena por lo que será en extremo útil para crear una mejor experiencia.

2.3. Trabajo Futuro

Para las próximas sesiones se plantea buscar nuevas formas de interacción e implementarlas, intentar implementar Go-Go en alguna o todas sus variantes, además de buscar algunas escenas complejas en internet para integrarlas al proyecto y comenzar a crear inmersión.

Idea: Crear una escena de una bolera. Así se podrían aprovechar las interacciones que se han implementado en esta sesión con lanzamiento de bolas y derribar objetos verticales.

Idea: Basar el trabajo en un pequeño minijuego deportivo inmersivo (bolos, baloncesto, etc...).

3. Día 3: 24/06/2017

3.1. Hechos

Se cambia la interacción del usuario con el entorno al forzarle a utilizar un gesto para desplazarse, a saber, apuntar con el dedo índice. Esto se ve reforzado con el Avatar SDK ya que se pueden ver exactamente los movimientos que están haciendo las manos.

Se integran "Mallas de Navegación"(Nav Meshes) así como agentes de navegación a una escena sencilla para que la navegación se realice utilizando algoritmos de *pathfinding*.

Se comienza a experimentar con escenas complejas.

3.2. Observaciones

Al cambiar la interacción se le da al usuario mayor juego a la hora de interactuar con la escena ya que, al no tener siempre la esfera que muestra el destino, se consigue una mayor inmersión.

Utilizando las mallas de navegación conseguimos un movimiento mucho más natural por parte del usuario, evitando así *atravesar* objetos del escenario, lo cual, sacaría al usuario de la experiencia inmersiva.

Integrar escenas complejas supone un nuevo problema de navegación ya que hay que adaptar las mallas de navegación a cada parte de la escena.

También se han detectado problemas dependiendo del modelo que se utiliza por lo que se asume que el problema lo tiene el modelo en sí y no la implementación.

3.3. Trabajo Futuro

Adaptar algunas escenas complejas a la navegación con mallas de navegación para que resulte más inmersivo para el usuario.

Cambiar la estética del objeto de destino, en vez de utilizar una bola, utilizar un emisor de partículas.

Comenzar a integrar objetos interactivables a las escenas complejas.

4. Día 4: 27/06/2017

4.1. Hechos

Se ha adaptado el script de movimiento para funcionar mejor con escenas complejas. Creada escena compleja utilizando Avatar SDK y NewtonVR

para representar al usuario, dando como resultado un entorno inmersivo y complejo.

Se ha adaptado el script de movimiento para adaptar la dirección en la que mira el usuario cuando se mueve para reducir la sensación de desorientación.

Se ha probado a cambiar la esfera de destino por un emisor de partículas.

Creado un panel para informar al usuario que tipo de movimiento está empleando.

4.2. Observaciones

Integrando lo ya visto antes de NewtonVR y el Avatar SDK a una escena compleja se obtiene como resultado una escena sorprendentemente inmersiva por si misma.

La adaptación de la dirección para disminuir la desorientación al desplazarse puede llegar a causar mareo. Habría que dar la opción de quitarlo o hacer una encuesta en la que se pregunté sobre que tipo de desplazamiento se prefiere.

El cambio de la esfera al emisor de partículas ha dado buen resultado visual. No obstante habría que reforzar más este indicador ya que: el emisor de partículas no está adecuadamente configurado y hace falta algo más para dar bien el feedback al usuario de en dónde va a aparecer o hacia donde va a desplazarse. En lo referente al desplazamiento se prefiere precisión antes que inmersión.

El panel, aunque supone un elemento extradiegético que saca al usuario un poco de la experiencia, se considera necesario para que el usuario no se "pierda".^{en} cuanto al tipo de movimiento que va a utilizar.

Idea: Usar algún elemento intradiegético para mostrar esta información, tal como un reloj en el brazo, alguna interfaz futurista o incluso cambiar el elemento visual que nos dice el tipo de movimiento por un elemento auditivo.

4.3. Trabajo Futuro

Trabajar en la integración de sonidos a la escena para aumentar la inmersión. Cambiar algunos elementos informativos de extradiegéticos a intradiegéticos para que la experiencia sea más inmersiva.

Implementar la navegación mediante pathfinding en escenarios complejos. Investigar técnicas para reducir el mareo aparte del enfoque dinámico que se puedan aplicar a este proyecto.

Comenzar a implementar scripts similares en Unreal Engine para ir teniendo algo sobre lo que trabajar en este motor.

5. Día 5: 28/06/2017

5.1. Hechos

Cambiado el script de control con raycast y divididos los movimientos en distintas clases aplicando así el patrón de diseño **Estrategia**.

5.2. Observaciones

Al implementar el movimiento a partir del patrón estrategia se aumenta la capacidad de modificación del script de control y se mejora la claridad del código.

5.3. Trabajo Futuro

Sin cambios en cuanto al trabajo futuro de los días anteriores.

6. Día 6: 29/06/2017

6.1. Hechos

Tras estudiar el artículo [Davis et al.] y viendo las conclusiones del mismo se aprecia que surgen menos mareos en escenas menos complejas.

Debido a esto se incorpora un nuevo modelo para hacer pruebas. Este nuevo modelo está formado a partir de texturas mucho más sencillas que el anterior modelo por lo que se espera que el uso del mismo reduzca notablemente el posible mareo que puedan tener los futuros usuarios.

Se ha implementado un sistema de profundidad de campo dinámica similar a la mostrada en el artículo [Carnegie and Rhee].

6.2. Observaciones

El nuevo modelo más simple supone una carga computacional y de renderizado bastante menor que los otros modelos más complejos por lo que, aparte de suponer un factor de reducción de la fatiga a partir del uso de modelos menos complejos, se verá incrementada la velocidad de la escena por lo que el usuario sentirá aún menos fatiga.

El nuevo sistema de enfoque dinámico implementado a partir de la profundidad de campo dinámica se comporta de una manera mucho mejor que el anterior sistema. Con un desenfoque de mayor calidad y un enfoque mucho más efectivo y suave que antes.

6.3. Trabajo Futuro

Ajustar el sistema de enfoque dinámico para que los valores de blur no sean demasiado molestos para el usuario.

Probar la nueva escena menos compleja y adaptar las mallas de navegación para la escena con "jugador usando VR HMD".

Referencias

- [Porcino et al.] THIAGO M. PORCINO, ESTEBAN CLUA, DANIELA TREVISAN, CRISTINA N. VASCONCELOS AND LUIS VALENTE
Minimizing cyber sickness in head mounted display systems: design guidelines and applications, Fluminense Federal University, Institute of Computing, 2016
- [Davis et al.] SIMON DAVIS, KEITH NESBITT AND EUGENE NALIVAIKO
Comparing the onset of cybersickness using the Oculus Rift and two virtual roller coasters, Proceedings of the 11th Australasian Conference on Interactive Entertainment (IE 2015), vol. 27, 2015, p. 30.
- [Carnegie and Rhee] KIERAN CARNEGIE AND TAEHYUN RHEE
Reducing Visual Discomfort with HMDs Using Dynamic Depth of Field, Computer Graphics and Applications, IEEE, vol. 35, no. 5, pp. 34–41, 2015.