**INFORME AVANCE NAVAR QA**

1. **Objetivos principales:**

* Pruebas y toma de datos con Software de tracking Motive para mejoramiento de la precisión del Pointer.
* Mejoramiento del rastreo de la broca y disminución de ruido en el entorno Unity.
* Método para registrar broca con teclado.
* Implementación de configuración de cámara en motive.

1. **Descripción del proceso:**
2. **Pruebas y toma de datos con software de tracking Motive para mejoramiento de la precisión del Pointer.**

**Objetivo**

Realizar diferentes pruebas para mejorar la precisión del Pointer.

**Procedimiento:**

Se realizaron las siguientes pruebas, algunas se repitieron debido a que salen datos corruptos, sin embargo, se mencionan aquellas que los datos salieron correctos.

1. **Pointer punto**

Se posiciona el pointer en todo el centro del Phantom, se gira en diferentes sentidos y se toman setenta mil datos.

1. **Pointer inclinado**

Se ubica el pointer en tres posiciones: centrado, inclinado a la izquierda e inclinado a la derecha, se repite el procedimiento en tres puntos en el eje z: a 60 cm, 90 cm y 120 cm. Se toman treinta mil datos en cada posición.

1. **Barra rotada 180°**

Se ubica la barra en un punto central de visión de las cámaras y se rota 180°, se repite el procedimiento en tres puntos en el eje z: a 60 cm, 90 cm y 120 cm. Se toman setenta mil datos en cada posición.

1. **Pointer Rotado**

Se ubica el pointer en el punto central del Phantom y se rota de izquierda a derecha en tres puntos en el eje z: a 60 cm, 90 cm y 120 cm.

**Resultado**

|  |  |  |
| --- | --- | --- |
| **Nombre Prueba** | **Datos** | **Error XY** |
| **Pointer Punto** | **70000** | **0.6 mm** |
| **Pointer Inclinado** | **90000** | **5 mm** |
| **Barra rotada 180°** | **3000** | **0.9 mm** |
| **Pointer rotado** | **3000** | **0.4 mm** |

**Conclusiones**

* Se logró obtener el mejor proceso de calibración y configuración de las cámaras para obtener un error aceptable.
* Se concluye cambiar el pointer por uno que tenga menor distancia de la punta a la primera esfera para disminuir aún más el error.

1. **Mejoramiento del rastreo de la broca y disminución de ruido en el entorno Unity.**

**Objetivo:**

Disminuir el ruido del instrumento broca que se produce al rotarla con taladro.

**Procedimiento**

1. **Crear método que promedie datos de entrada de motive** 
   1. Crear lista que acumule posiciones según un contador.
   2. Al finalizar el contador realizar el promedio y asignar a la conversión de quaterniones a ángulos de Euler hecho por Unity.
2. **Crear método para dar rigidez a la broca luego de asignar datos de entrada al objeto.**
   1. Crear una variable de weight o peso que al aumentar o disminuir el valor suceda lo mismo con el ruido.
   2. Promediar entre dos entradas.

**Resultado**

1. **Método para promediar entradas de Motive**

void UpdatePose**()** **{**

OptitrackRigidBodyState rbState **=** StreamingClient**.**GetLatestRigidBodyState**(**RigidBodyId**);**

**if** **(**rbState **!=** **null)** **{**

**if** **(this.**RigidBodyId **==** 5**)** //promediar broca

**{**

**if** **(**counter **<** counterLimit**)** **{**

//acum = rbState.Pose.Position;

listAcum**.**Add**(**rbState**.**Pose**.**Position**);**

counter**++;**

**}** **else** **{**

var newVector **=** **new** Vector3**(**

listAcum**.**Average**(**x **=>** x**.**x**),**

listAcum**.**Average**(**x **=>** x**.**y**),**

listAcum**.**Average**(**x **=>** x**.**z**));**

**this.**transform**.**localPosition **=** newVector**;**

**this.**transform**.**localRotation **=** rbState**.**Pose**.**Orientation**;**

counter **=** 0**;**

**}**

**}** **else** **{**

**this.**transform**.**localPosition **=** rbState**.**Pose**.**Position**;**

**this.**transform**.**localRotation **=** rbState**.**Pose**.**Orientation**;**

**}**

**}**

**}**

1. **Método para dar rigidez a la broca luego de asignar datos de entrada al objeto.**

**[**Range**(**0.0f**,** 1.0f**)]**

**public** float weight **=** 0.5f**;**

**Vector3** old\_forward3**,** old\_right3**;**

bool firstFrame **=** **true;**

void Update**(){**

**if** **(**firstFrame**)**

**{**

// Guarda el primer frame

old\_forward3 **=** obj2**.**transform**.**forward**;**

old\_right3 **=** obj2**.**transform**.**right**;**

firstFrame **=** **false;**

**}**

**else**

**{**

//La orientación de obj3 es un promedio ponderado de la orientación actual de obj2 y la orientación anterior de obj3

**Vector3** new\_forward3 **=** **((**weight**)** **\*** obj2**.**transform**.**forward **+** **(**1 **-** weight**)** **\*** old\_forward3**).**normalized**;**

**Vector3** new\_right3 **=** **((**weight**)** **\*** obj2**.**transform**.**right **+** **(**1 **-** weight**)** **\*** old\_right3**).**normalized**;**

//Use el producto cruz para asegurarse de que new\_up3 sea perpendicular a new\_forward3.

**Vector3** new\_up3 **=** Vector3**.**Cross**(**new\_forward3**,** new\_right3**).**normalized**;**

obj3**.**transform**.**rotation **=** Quaternion**.**LookRotation**(**new\_forward3**,** new\_up3**);**

old\_forward3 **=** obj3**.**transform**.**forward**;**

old\_right3 **=** obj3**.**transform**.**right**;**//

**}**

**}**

**Conclusiones**

Se observa una mejora considerable en el ruido y saltos de la broca al taladrar dentro del objeto en Unity.

1. **Método para registrar broca con teclado**

**Objetivo**

Añadir controles para manipular la broca y registrarla manualmente.

**Procedimiento**

Asignar 1,2,3 y 5 en el KeyPad para las direcciones en x e y de la broca.

**Resultado**

void Update()

{

float count=0.05f;

if (Input.GetKeyDown(KeyCode.Keypad5)) //eje y

{

Vector3 positiony = obj4.transform.position;

positiony.y+=count;

obj4.transform.position = positiony;

}

if (Input.GetKeyDown(KeyCode.Keypad2)) //eje -y

{

Vector3 positiony = obj4.transform.position;

positiony.y -= count;

obj4.transform.position = positiony;

}

if (Input.GetKeyDown(KeyCode.Keypad1)) //eje x

{

Vector3 positiony = obj4.transform.position;

positiony.x += count;

obj4.transform.position = positiony;

}

if (Input.GetKeyDown(KeyCode.Keypad3)) //eje -x

{

Vector3 positiony = obj4.transform.position;

positiony.x -= count;

obj4.transform.position = positiony;

}

}

**Conclusiones**

El sistema puede manipular la broca con el teclado en caso de que se desee registrar manualmente.

1. **Implementación de configuración de cámara en motive.**

**Objetivo**

Añadir sliders o barras de deslizamiento para configurar parámetros de las cámaras.

**Procedimiento**

Añadir conjunto de Widgets y aplicar layout en QtDesigner para actualizar la interfaz de Usuario

Realizar update para que actualice los sliders.

**Resultado**

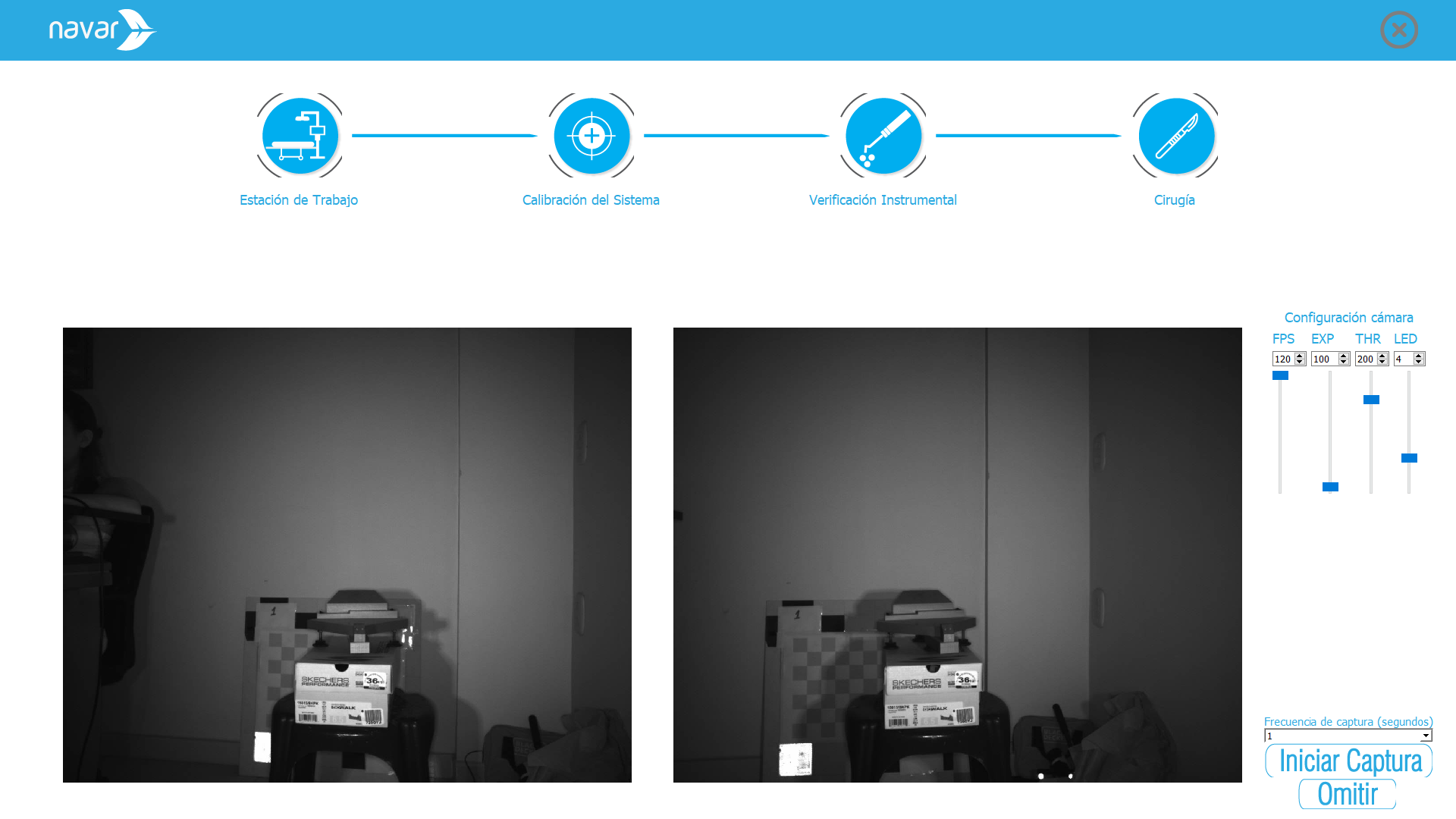


Ilustración 1 Ventana completa Navar con sliders añadidos.

**Conclusiones**

El sistema permite manipular parámetros de las cámaras en su interfaz de usuario.