Máster en Programación de Videojuegos
Fundamentos Matemáticos
Profesor José María Benito

Práctica 3: Cuaternios

Introducción

El objetivo de esta práctica es obtener una primera impresión de las posibilidades y limitaciones de de las operaciones con cuaterniones

Notas Generales

Esta práctica está diseñada para ser realizada en clase y no va a ser corregida o calificada. Sin embargo debes intentar completar todos los puntos y aprovechar el tiempo en el aula. En caso contrario será motivo para suspender la asignatura.

Creación del proyecto en Unity

En esta práctica vas a crear un proyecto de Unity en el que experimentar con rotaciones mediante cuaternios. La idea principal es rotar un personaje para poder observarlo desde distintas posiciones: frente, espalda, izquierda, derecha, arriba y abajo.

1. En Unity:
   1.1. Crea un proyecto sin importar ninguna asset. Dale un nombre tipo “quaternions”. Salva la escena dentro del directorio del proyecto.
   1.2. Añádele a la escena una cápsula (GameObject >> Create Other >> Capsule) para el cuerpo del personaje. Llama a este objeto “Body”.
   1.3. Crea dos cápsulas dispuestas simétricamente a ambos lados del cuerpo, de tamaño similar a unos brazos, y llámalas “armL" y “armR”. Utiliza el inspector para rotar y posicionar. Después haz los brazos hijos jerárquicos de “Body” (en pestaña Hierarchy, arrastra los brazos sobre body). De esta manera los brazos se moverán y rotarán con “Body”.
   1.4. Añade una esfera (GameObject >> Create Other >> Sphere) para la cabeza. Esta también debe colgar jerárquicamente de Body.

1.5. Añade una cámara apuntando directamente a “Body” y un par de luces para iluminar.

1.6. Añade un script de código a la cámara:

1.6.1. En Hierarchy, clica sobre Main Camera.

1.6.2. En el Inspector (Main Camera) >> Add Component >> New Script y dale un nombre tipo MainCameraScript.

1.7. Añade un script de código a Body. Repite los pasos anteriores y llámalo BodyScript.

1.8. Para diferenciar la parte anterior o posterior del cuerpo, añádele una textura a la cabeza.

1.8.1. Busca la imagen Smilie en el entregable.

1.8.2. Añade un material Diffuse o cambia el material de la cabeza a Diffuse. En la imagen asociada, elige la imagen Smilie.png.

2. En MonoDevelop (Unity >> Assets >> Sync Mlnodevelop Project)

2.1. Añade un interfaz de usuario de seis botones para las seis posiciones del cuerpo: Front, Left, Right, Top, Bottom, Back. Añade la siguiente función a MainCameraScript:

```
void OnGUI () {
            // Make a background box
            GUI.Box(new Rect(10,10,100,200), "Views");


            if(GUI.Button(new Rect(20,40,80,20), "Front")) {
                  bodyController.changeView("Front");
            }

            if(GUI.Button(new Rect(20,70,80,20), "Left")) {
                  bodyController.changeView("Left");
            }

            if(GUI.Button(new Rect(20,100,80,20), "Right")) {
                  bodyController.changeView("Right");
            }

            if(GUI.Button(new Rect(20,130,80,20), "Top")) {
                  bodyController.changeView("Top");
            }

            if(GUI.Button(new Rect(20,160,80,20), "Bottom")) {
                  bodyController.changeView("Bottom");
            }

            if(GUI.Button(new Rect(20,190,80,20), "Back")) {
                  bodyController.changeView("Back");
            }

      }
```

2.2. Añade la siguiente variable miembro a MainCameraScript.

```
public BodyScript bodyController;
```

Guarda el fichero MainCameraScript.cs

2.3. Para que la variable bodyController anterior esté inicializada adecuadamente al comienzo del programa, debes ir al editor de Unity;

2.3.1. Hierarchy >> MainCamera

2.3.2. Clica y arrastra desde Hierarchy el objeto “Body” al zócalo de “Body Controller” de “Main Camera” en el Inspector.

2.4. Escribe las siguientes funciones en BodyScript:

```
public void changeView( string view )
	{
		if ( view == currentView)
			return;


		object[] parms;
		currentView = view;
		StopCoroutine( "RotateToView" );

		switch ( view )
		{
		case "Left":
			parms = new object [2] {Vector3.left,Vector3.up};
			StartCoroutine( "RotateToView", parms );
			break;

		case "Front":
			// TODO
			break;

		case “Top":
			// TODO
			break;

		case "Bottom":
			// TODO
			break;

		case "Right":
			// TODO
			break;

		case "Back":
			// TODO
			break;

		default:
			break;
		}
	}

private IEnumerator RotateToView( object[] parms ) {
	Vector3 direction = (Vector3) parms[0];
	Vector3 up = (Vector3) parms[1];
```

```
        // TODO
}
```

En la función RotateToView debes hacer que el personaje alcance la orientación correspondiente a que su vector transform.forward apunte al vector pasado como primer parámetro, y su vector transform.up apunte al vector pasado como segundo parámetro. Además la rotación debe ser gradual. Primero define los vectores destino en changeView. Después intenta realizar la reorientación de golpe, aplicando a transform.orientation la siguiente rotación:

```
Quaternion rotation=
Quaternion.FromToRotation(currentVector,targetVector);
transform.rotation = rotation*orientation;
```

Para rotar de forma gradual puedes utilizar, por ejemplo:

```
Quaternion.Lerp
```

si la rotación es pequeña en cada paso, o la función

```
Quaternion.Slerp
```

si la rotación es amplia. Para que el programa siga respondiendo mientras se realiza la reorientación, la función RotateToView utiliza un método poco estándar para ejecutarse en "segundo plano" de la siguiente forma:

```
private IEnumerator RotateToView( object[] parms ) {

        // …

        // done es una variable booleana que controla el bucle

        while (! done) {

                // pon tu código de rotación gradual aquí

                // la siguiente instrucción le dice al runtime que ceda
                // la ejecución a otros procesos para no bloquear el hilo
                // principal, pero que debe volver al bucle while
                // en el siguiente frame
                yield return null;
        }

        // rotación ha terminado (bucle while ha finalizado)
        yield return "done";
}
```

Cualquier método que se te ocurra es adecuado. Este es un problema abierto para experimentar así que no hay una solución estándar, usa la imaginación. Si tienes demasiadas dudas pregunta al profesor para recibir alguna pista.

Si tienes tiempo extra y quieres experimentar más, utiliza el objeto “Input” de Unity para obtener los movimientos y botones del ratón, y usa cuaternios para rotar el personaje según el movimiento de ratón del usuario.