

UNIVERSIDAD PANAMERICANA

Reporte de resultados y procedimientos

NOMBRES: Aranda Mejía Brian Antonio 0209486
Pineda Cervantes Adrián 0199347
San Román Hernández José Manuel 0204799

PROFESOR: Dr. David Escobar Castillejos

ESPECIALIDAD: Simuladores Gráficos

MATERIA: Dispositivos Hapticos

FECHA: 22/03/2023

Universidad Panamericana

Facultad de Ingeniería

Introducción

A lo largo del curso, hemos profundizado en el fascinante mundo de la tecnología háptica, explorando tanto sus fundamentos teóricos como prácticos. Para poder desarrollar una simulación efectiva, fue necesario también adquirir conocimientos sobre los principios físicos subyacentes al fenómeno a representar. En este proceso, tuvimos que descartar temas de alta complejidad que no eran relevantes para nuestro proyecto. Utilizando la tecnología háptica, que se enfoca en el estudio del sentido del tacto, y la ley de Hooke para modelar la elasticidad de un resorte, pudimos crear una simulación precisa de la interacción física involucrada en el tiro con arco.

Simulación

El objetivo de este proyecto consiste en la simulación de la interacción física que existe en la disciplina del tiro con arco mediante un dispositivo háptico Novint-Falcon. Mediante este se buscó asimilar el movimiento que da dirección al tiro, y los cambios de fuerza al tensar y soltar la cuerda del arco haciendo uso de los botones y componentes móviles del dispositivo.

La simulación consiste en un campo de tiro con arco con objetivos móviles y viento ambiental, y un arco conectado al dispositivo háptico. El movimiento del dispositivo háptico mueve el arco alrededor del campo y presionar el botón cargará el arco para disparar una flecha, en medio de cargar la flecha el usuario puede tirar del dispositivo háptico y sentir la tensión que se acumula en la cuerda. Al soltar el botón se dispara la flecha y da un pequeño golpe al usuario.

Proceso

Inicialmente buscamos assets y ambientes para darle una espacialidad a los objetos a introducir, se realizó una primera simulación utilizando el mouse como el dispositivo que mueve el arco t click izquierdo como el tensor de la cuerda, mientras más tiempo se dejaba apretado, más fuerte disparaba la flecha.

El siguiente paso fue adaptar el movimiento del mouse al del dispositivo háptico, en este punto tuvimos inconvenientes dentro de la dirección del eje al mover el dispositivo dentro del terreno, tuvimos que girar 90° el terreno completo para que correspondiera con la dirección natural del dispositivo, así como el arco para que apuntará hacia la dirección correcta.

El siguiente problema surgió al adaptar el click del mouse al click del botón del háptico, existían errores dentro del código que no nos permitía activar el botón y separar el temporizador para darle el movimiento de los ejes, entonces se separaron en dos, un botón sujetaba la acción y al jalar el dispositivo se siente la tensión de la

cuerda, finalmente con el segundo botón se libera la cuerda, lanzando la flecha.

Junto con este problema encontramos que el arco disparaba aún sobre la dirección original y que la fuerza no se modifica al sostener la cuerda; la solución de el primero se tuvo que redireccionar la salida de la flecha y para el segundo se tuvo que crear un proceso de diferencia entre los ejes del dispositivo háptico.

Consideraciones

Lo que tomamos en cuenta en principio fueron todos los asset y el ambiente de la simulación, la escena arenosa, partículas de polvo y arena moviéndose para el efecto, tiros al blanco en movimiento y cajas, barriles y palmeras para darle una dificultad al disparar; adicionalmente el asset del arco y la flecha que aparece al ser disparada, junto con el sonido de disparo.

Dentro de las fuerzas, tomamos en cuenta el limitante de la cuerda y su posición para sacar disparada la flecha en cuanto a la distancia tensada de la cuerda, utilizando la fuerza de 2000f originaria del programa. Dicha fuerza resulta también la velocidad con la que la flecha sale del arco.

Las interacciones son de la flecha con los tiros al blanco, al momento de hacer contacto el objetivo se destroza, al igual que los barriles y cajas, al momento del contacto éstos se pueden mover como efecto secundario.

Resultados

La simulación dentro del entorno correcto del terreno funciona correctamente, el arco se moviliza dentro del espacio delimitado para el tiro con arco y los objetos coinciden correctamente con los ejes denominados del dispositivo háptico, inclusive la flecha dispara correctamente bajo el eje perpendicular al arco.

El disparo de la flecha se logró delimitar a una carga anterior al disparo aunque éste fue desarrollado en dos botones, uno para detener el arco y permitir la carga de la tensión de la cuerda y el otro para soltar la carga, disparando la flecha.

De igual manera además de sentir la tensión de la cuerda al cargar la flecha, al soltar la misma se obtiene una sensación de retroceso en impacto.

Fotos

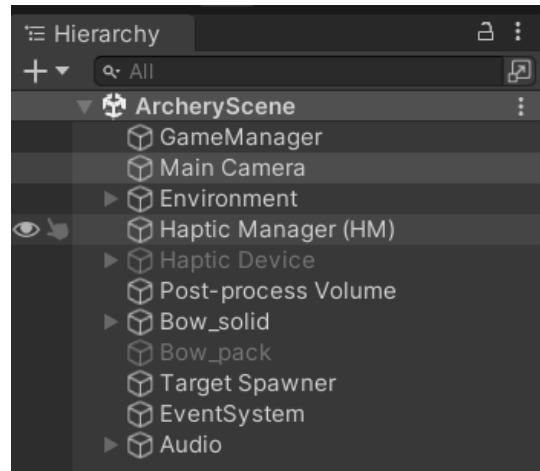


Imagen 1: Jerarquía de los objetos utilizados dentro de la escena.

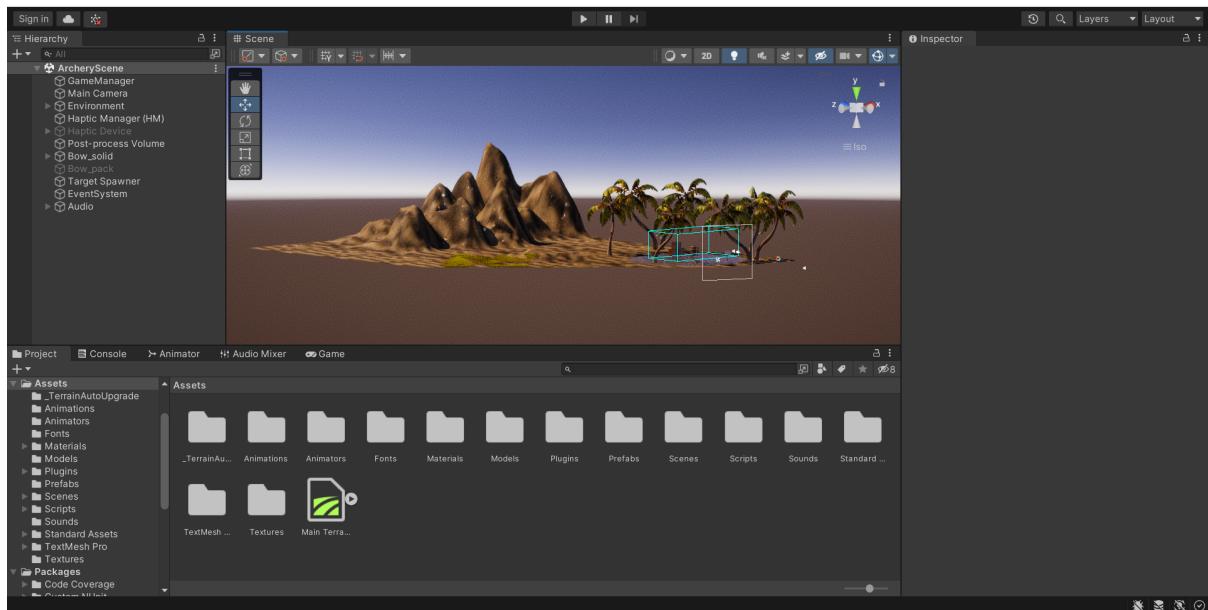


Imagen 2: Visualización de la escena junto a los asset utilizados dentro del entorno.

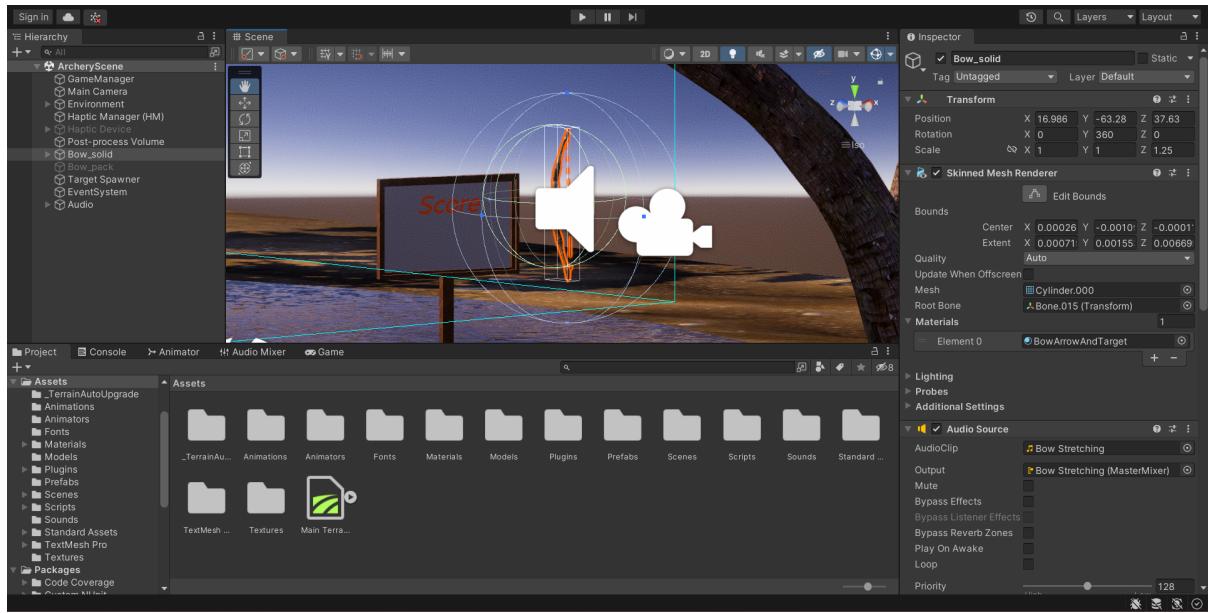


Imagen 3: Visualización del arco unido al dispositivo háptico y sus características dentro del entorno.

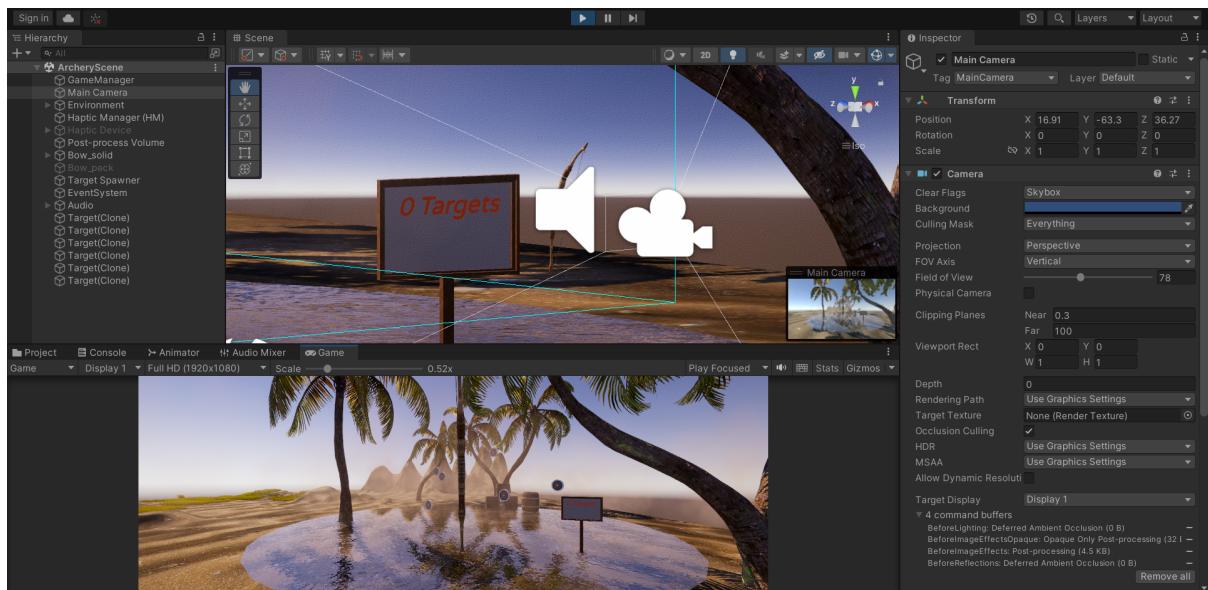


Imagen 4: Simulación del proyecto en tiempo real

Brevemente anexamos a continuación imágenes del código utilizado para emparejar el dispositivo háptico con los objetos a interactuar en la escena:

HapticManager.cs

```
void Update ()
{
    arrowTransform.rotation = Quaternion.Euler(180, 180, 180);
    if (pushButton0 == true && arrowIsInstantiated == false)
    {
        arrowIsInstantiated = true;
        pushButton3 = false;
        pushButton0 = false;
        bowAnimator.SetBool("drawing", false);
        arrow = Instantiate(arrowPrefab, arrowTransform.position, arrowTransform.rotation) as GameObject;
        arrow.GetComponent<Rigidbody>().AddForce(arrowTransform.forward * currentArrowForce);
        currentArrowForce = 0;
        currentLerpTime = 0;
        Debug.Log(arrowTransform.rotation);
    }
    if (pushButton3 == true)
    {
        timePress += Time.deltaTime;
        bowAnimator.SetBool("drawing", true);
        if (bowAnimator.GetBool("drawing"))
        {
            currentLerpTime += Time.deltaTime;
            if (currentLerpTime > maxLerpTime)
            {
                currentLerpTime = maxLerpTime;
            }
            float perc = currentLerpTime / maxLerpTime;
            currentArrowForce = Mathf.Lerp(0, maximumArrowForce, perc);
        }
    }
}
```

```
    else
    {
        timePress = 0;
    }
    if (arrow == null)
    {
        //timePress = 0;
        arrowIsInstantiated = false;
    }
    // Exit application
    if (Input.GetKey(KeyCode.Escape))
    {
        Application.Quit();
    }
}
```

```

void HapticThread()
{
    while (hapticThreadIsRunning)
    {
        for (int i = 0; i < hapticDevices; i++)
        {
            // get haptic positions and convert them into scene positions
            position[i] = workspace * HapticPluginImport.GetHapticsPositions(myHapticPlugin, i);
            orientation[i] = HapticPluginImport.GetHapticsOrientations(myHapticPlugin, i);

            // get haptic buttons
            button0[i] = HapticPluginImport.GetHapticsButtons(myHapticPlugin, i, 1);
            button1[i] = HapticPluginImport.GetHapticsButtons(myHapticPlugin, i, 2);
            button2[i] = HapticPluginImport.GetHapticsButtons(myHapticPlugin, i, 3);
            button3[i] = HapticPluginImport.GetHapticsButtons(myHapticPlugin, i, 4);

            if (button0[i])
            {
                pushButton0 = true;
                Vector3 ejeZHaptic = new Vector3(0, 0, 5);
                HapticPluginImport.SetHapticsForce(myHapticPlugin, i, ejeZHaptic);
            }

            if (button3[i])
            {
                pushButton3 = true;
                Vector3 ejeZHaptic = new Vector3(0, 0, timePress * 5);
                HapticPluginImport.SetHapticsForce(myHapticPlugin, i, ejeZHaptic);
                Debug.Log(timePress);
            }
            else
            {
                pushButton3 = false;
            }
        }

        HapticPluginImport.UpdateHapticDevices(myHapticPlugin, i);
    }
}

```



Imagen 5: Visualización del funcionamiento del programa en conjunto al dispositivo háptico.

Reflexiones

El simulador podría mejorar en el aspecto de funcionalidad bajo la utilización de un sólo botón para realizar la carga y descarga de la flecha, adicionalmente se puede mejorar la programación de la sensación de la tensión de la cuerda gradualmente utilizando la posición del punto de carga a comparación de su punto original en las coordenadas de la flecha y así poder obtener de la misma manera el retraer la cuerda y no disparar la flecha una vez cargada. Otra cosa a mejorar es en la animación ya que al momento de realizar la animación de querer tensar la cuerda, este no va acorde cuando el usuario mueve el háptico hacia él debido a que está programado que se ejecute cuando el usuario presiona el primer botón para generar la fuerza con la cual va a salir disparada la flecha.

Por otra parte, una de las mayores dificultades que tuvimos al realizar el proyecto fue el no poder encontrar documentación apropiada en internet para el uso del Háptico Noivnt Falcon y el tener conocimientos básicos en Unity ya que al ser herramientas nuevas o poco usadas, no teníamos idea de cómo plasmar la acción que queríamos que se realizará con el háptico en Unity pero al final se logró un resultado favorable. La única ayuda que se nos proporcionó fue el proyecto que realizó el profesor para que el háptico funcionara de manera correcta en Unity pero nosotros al no tener conocimiento alguno del proyecto y del código nos fue muy difícil entenderlo pero conforme pasaron los días y le preguntamos al profesor las dudas se fueron resolviendo, por ejemplo, al mover el háptico en el espacio de Unity, este se movía de manera muy brusca y para solucionarlo solo teníamos que modificar el workspace.