

Documentación de la Clase Inputs

Gerstep

August 28, 2024

1 Introducción

La clase `Inputs` es responsable de gestionar las entradas del usuario en un entorno gráfico 3D utilizando `OpenTK`. Esta clase captura las entradas del teclado y del ratón, y las traduce en acciones que controlan el movimiento y la orientación de la cámara en la escena. También permite capturar pantallas y manejar el estado del cursor.

Esta clase es parte del espacio de nombres `OpenTK.Tarea_3.Controlador` y proporciona la lógica necesaria para manejar interacciones del usuario en un entorno gráfico.

2 Atributos Privados de la Clase

2.1 `_camera`

- **Tipo de Dato:** `camera`
- **Descripción:** Una referencia a la instancia de la clase `camera`, que controla la posición, orientación y movimiento de la cámara en la escena 3D. Las entradas del usuario modifican esta cámara.

2.2 `_window`

- **Tipo de Dato:** `GameWindow`
- **Descripción:** Una referencia a la ventana de juego (`GameWindow`) que se utiliza para capturar el estado del teclado y del ratón, así como para manejar la interfaz gráfica de la aplicación.

3 Constructor de la Clase

```
public Inputs(camera camera, GameWindow window)
```

3.1 Descripción

El constructor de la clase **Inputs** inicializa las referencias a la cámara y la ventana del juego, preparándolas para gestionar las entradas del usuario.

3.2 Parámetros

- **camera:** Una instancia de la clase **camera**, que se encargará de procesar las entradas del usuario para mover y orientar la cámara.
- **window:** Una instancia de **GameWindow**, que proporciona acceso a los estados del teclado y del ratón, así como a otras funciones relacionadas con la ventana de la aplicación.

3.3 Funcionamiento

El constructor realiza las siguientes acciones:

1. Asigna la referencia de la **camera** pasada como parámetro al atributo privado **_camera**.
2. Asigna la referencia de la **window** pasada como parámetro al atributo privado **_window**.

3.4 Ejemplo de Uso

```
camera camara = new camera(new Vector3(0.0f, 0.0f, 3.0f));
GameWindow ventana = new GameWindow(GameWindowSettings.Default, NativeWindowSettings.Default);
Inputs entradas = new Inputs(camara, ventana);
```

```
// Esto inicializa la gestión de entradas para la cámara y la ventana del juego.
```

4 Métodos de la Clase

4.1 HandleInput (Gestionar la Entrada del Usuario)

```
public void HandleInput(FrameEventArgs e)
```

4.1.1 Descripción

Este método gestiona las entradas del usuario en cada frame de la aplicación. Captura las entradas del teclado y del ratón, y las traduce en acciones que afectan el movimiento y la orientación de la cámara. También permite realizar capturas de pantalla, bloquear o liberar el cursor, y cerrar la ventana.

4.1.2 Parámetro

- **e**: Un objeto de tipo `FrameEventArgs` que contiene información sobre el tiempo transcurrido desde el último frame, utilizado para asegurar que el movimiento de la cámara sea independiente de la velocidad de fotogramas.

4.1.3 Funcionamiento

El método `HandleInput` sigue los siguientes pasos:

1. ****Movimiento de la cámara con teclas:****
 - Inicializa un vector `direction` en `Vector3.Zero`.
 - Captura las teclas presionadas utilizando el estado del teclado proporcionado por `_window.KeyboardState`:
 - **W**: Mueve la cámara hacia adelante, incrementando la componente *Z* del vector `direction`.
 - **S**: Mueve la cámara hacia atrás, decrementando la componente *Z* del vector `direction`.
 - **A**: Mueve la cámara hacia la izquierda, decrementando la componente *X* del vector `direction`.
 - **D**: Mueve la cámara hacia la derecha, incrementando la componente *X* del vector `direction`.
 - **LeftControl**: Mueve la cámara hacia abajo, decrementando la componente *Y* del vector `direction`.
 - **LeftShift**: Mueve la cámara hacia arriba, incrementando la componente *Y* del vector `direction`.
 - **F**: Toma una captura de pantalla utilizando `Screenshot.SaveScreenshot` y muestra un mensaje en la consola.
 - **Q**: Bloquea el cursor dentro de la ventana y permite que la cámara responda al movimiento del ratón (`bloquear_raton = false`).
 - **E**: Libera el cursor, haciendo que la cámara deje de responder al movimiento del ratón (`bloquear_raton = true`).
 - **Escape**: Cierra la ventana de la aplicación.
2. ****Aplicar el movimiento a la cámara:****
 - Llama al método `_camera.ProcessKeyboardInput`, pasando el vector `direction` y el tiempo transcurrido (`e.Time`) para mover la cámara en la dirección deseada.
3. ****Movimiento de la cámara con el mouse:****
 - Captura el estado del ratón utilizando `_window.MouseState`.
 - Llama al método `_camera.ProcessMouseMovement`, pasando los desplazamientos *X* y *Y* (`mouseState.Delta.X` y `mouseState.Delta.Y`) para ajustar la orientación de la cámara.

4.1.4 Ejemplo de Uso

```
// Este método se llamará en cada frame para manejar las entradas del usuario
ventana.UpdateFrame += (FrameEventArgs e) =>
{
    entradas.HandleInput(e);
};
```

5 Conclusión

La clase `Inputs` es crucial para gestionar las interacciones del usuario en aplicaciones de gráficos 3D con OpenTK. Al capturar las entradas del teclado y del ratón, permite mover y orientar la cámara en la escena, realizar capturas de pantalla, y controlar el estado del cursor. Esta clase proporciona una interfaz sencilla y eficaz para implementar controles interactivos en un entorno gráfico tridimensional.