**Clase 02 de Mayo**

Propiedad de instancia, valores locales

Valor de Up en Clase Vector3 determinada, valor único

Quaternion.identity = posición de rotación inicial

Movimientos:

1. Reposicionamiento
2. Vectorial
3. Trigonometria
4. Rotate.Around = utiliza librerías de métodos de trigonometría

No mezclar 2 tipos diferentes de movimientos (por ejemplo física y translate (reposicionamiento))

MoveTowards tiene validación de clampeo= toma un valor como límite, compensa la velocidad del movimiento, rectifica el movimiento

**Clase 07 de Mayo**

Top Down, Platformer

El if pueda considerar clases dentro del no

Extender la clase, se puede definir operaciones

Por defecto una de las operaciones tacitas que tienen casi todas las clases de Unity, operador de bool que tiene la clase por defecto

Error de runtime: uso de acceso que hacen referencia a null

Reglas de collision:

1. Ambos tienen que tener un collider
2. Al menos uno de los 2 debe tener el check de trigger
3. Al menos uno debe tener el método OnTrigger “ “
4. Por lo menos uno de los 2 debe tener el componente rigidbody

Polimorfismo

1. Puedes usar algo como algo mas
2. Puedes transformar al objeto en algo más dependiendo de la jerarquía

Casteo

**Clase 14 de Mayo**

**¿Cómo colisionar con sprites 2D?**

**¿Cómo generar una línea entre el puntero del mouse y un objeto (Gizmo)?**

Raytracing – colisión línea

La física no corre con cada frame, la física es simulada

Termina teniendo el mismo resultado visual que un bake

Procesos de luces dinámicas matan los celulares, los procesos de luces se hacen “horneado” “bake”

Iluminación precocida para juegos de consolas

Si el Clamp no se puede hacer directo se tiene que hacer dentro del método

**Dibujar el movimiento del mouse usando un Gizmo**

Dentro de un OnGUI, etiqueta con forma de rectángulo + Input.mousePosition

La posición del Mouse es como un Vector2

La posición del mouse se mide en pixeles en Unity

Se agrega Camera.main.ScreenToWorldPoint

Metodo Screen relación a pixeles de la pantalla

Agregar una variable Vector2 para almacenar la posición actual del mouse;

\*valor default the un float es 0.0

Nunca va a ser exacto la posición del gizmo porque dependerá de la capacidad del Engine

Gizmos Drawline del objeto a la posición del puntero del mouse

Utilizar la magnitud del vector que se dibuja desde el object hacia el mouse para mover la cámara

**Clase 16 de Mayo**

Vector que apunta con una magnitud y dirección. Punto de referencia es el transform mismo

El movimiento de la cámara se estuvo haciendo con Translate, pero no es conveniendte

No debe ser en base a perform position sino

Lo dibuja en función del punto de origen. El punto de origen de un vector es cero.

Si algo se pone en el inicio siempre se mantendrá la posición del inicio, get return obtiene un retorno cada frame.

GetMouseButton down para cuando el botón esta presionado, el numero 0 es el click izquierdo

Transform.position + playerDelta (uno es vector3 y el otro es VECTOR2 y no se pueden sumar)

Vector3 direction = dirección por defecto

**Clase 21 de Mayo**

El “While” se mantiene en un solo frame

CheckKey – MouseButton

MouseButton(0) = click izquierdo

MouseButton(1) = click derecho

Maneras de guardar información en variables

Time.Deltatime = diferencia de tiempo entre frame y frame

Contador en base de fracciones de tiempo;