

2.2. Corto documento de ejemplo

En la carpeta del motor, debe existir un documento que explique cómo crear un cuerpo usando el motor:

- Debes explicar cómo crear un cuerpo.
- Debes explicar cómo agregar el cuerpo a un world y que sucede si no se agrega al world.
- Debes explicar la función que utiliza la clase world para calcular las posiciones del cuerpo.
- Debes explicar cuál es la anatomía del frame y cómo afecta el tiempo de frame a la simulación.

Respuestas

1. Para crear un cuerpo, debemos instanciar un objeto de la clase body, en el caso de nuestro engine, sería vBody utilizando la sintaxis vBody [Nombre del Objeto].

Es importante recalcar que debemos tener un incluir la el world.h, en nuestro caso voidWorld.h utilizando #include.

voidWorld contiene una referencia a vBody por lo que podremos instanciar el objeto sin mayor dificultar.

2. Antes de agregar el cuerpo al world, debemos instanciar un world utilizando la sintaxis voidWorld [Nombre del objeto], en el cual podremos pasar por argumento un vec2D con el valor de la gravedad si así lo deseamos.

Una vez hagamos esto, podemos llamar la función del world instanciado llamada addBody(&vBody) y le pasamos por argumento el body antes instanciado.

Si no lo agregamos al world, simplemente no será tomado en cuenta en el momento de la simulación, por lo que no se le calculará velocidad ni posición.

3. Explicación de la función Step(float dt):

Esta función consta de dos iteradores 'for' que caminan a través del vector bodies para calcular en el primero la velocidad, y en el segundo la posición de los objetos dentro del vector.

En el primer 'for' se calcula la velocidad en 2 líneas, la primera línea maneja la coordenada en x, y la segunda línea maneja la coordenada en y.

Se utiliza el método de Euler para realizar los cálculos.

La ecuación nos dice que la velocidad es igual a la velocidad anterior sumada a la aceleración y multiplicado por delta tiempo.

Debemos tomar en cuenta que pueden existir aceleraciones y fuerzas que debemos de sumar, por lo tanto, debemos sumar la gravedad con la fuerza ejercida en el objeto y multiplicarlo por la más inversa del cuerpo y luego multiplicado por el deltatime.

Para calcular la aceleración se utiliza la formula $F=m*a$, de la cual al despejar la aceleración queda $a = F/m$, lo cual se puede también expresar como $a = F * 1/m$ (invMass)

En el segundo 'for' el cual calcula la nueva posición del cuerpo, es más sencillo, ya que la ecuación es mas corta. La nueva posición es igual a la posición anterior más la velocidad multiplicado por el deltatime.

Al multiplicar por deltatime, que es el tiempo de proceso de un frame, estaremos desvinculando la dependencia de los cálculos a cada frame y será con respecto al tiempo.

Es importante decir que la simulación será más exacta mientras el frametime sea mucho más bajo.

4. La anatomía del frame es simplemente de como cuales son las fases dentro del frame.

Se compone de 3 fases:

- En la primera se manejan los eventos, como por ejemplo el input del teclado.
- La segunda fase es el update, en la cual se puede calcular nuevas posiciones, velocidades y otros tipos de cálculos.
- La tercera fase es el render, en la cual se dibuja en pantalla todos los elementos deseados que componen la escena o el canvas.

Mientras más rápido se realicen estas 3 fases, mas bajo será el tiempo del frame, y mas fluida será la visualización.

Sin embargo, un tiempo de frame bajo permite realizar cálculos mas exactos en un motor de física basado en los métodos de Euler, ya que permitirá un mejor detalle en los movimientos de los elementos que estén dentro de la simulación, al aumentar el tiempo de render existirá una mayor interpolación en el movimiento del elemento, restando detalles visuales.

- **Ivan Gonzalez**
- **Logan Sarkar**