

Box 2D

Mikel Iturrlde Salguero

Animación 3D: Diseño y desarrollo de videojuegos

Descripción general	2
Contexto	3
Objetivos	3
Descripción técnica	3
Entidades simples	3
Circle	3
Cube	3
Edge	4
Entidades Complejas	4
Platform	4
Carretilla	4
Coche	4
Scene	4
Diario de desarrollo	5
Joints	5
Reflexiones	5
Posibles mejoras	5
Conocimientos adquiridos	5

Descripción general

Este proyecto consiste en la realización de un motor de una aplicación que utilice la librería box2D, la cual facilita enormemente la implementación de gráficos y físicas en dos dimensiones.

Contexto

Este proyecto se ha realizado como parte de una asignatura universitaria centrada en C++ como lenguaje principal. También se ha apoyado a nivel de código en ejemplos obtenidos del repositorio público: "<https://bitbucket.org/angel-esne/>".

Objetivos

El desarrollo de este proyecto cumple con los siguientes objetivos:

- Un vehículo empieza colocado en la parte superior de una rampa. El usuario controla el movimiento lateral del vehículo con el teclado (avance hacia la izquierda, hacia la derecha y freno).
- Se debe hacer descender el vehículo por la rampa con suficiente velocidad para que alcance el otro extremo.
- Cuando el vehículo llega al otro lado, la torreta rota su parte superior para dejar caer los círculos verdes sobre el vehículo y asciende una plataforma que tapa el agujero.
- Se hace avanzar el vehículo hasta la plataforma rosa.
- Cuando el vehículo se queda quieto sobre la plataforma, esta empieza a ascender hasta la altura de la plataforma flotante automáticamente.
- Se hace avanzar el vehículo hasta la plataforma flotante y, cuando está cerca del borde, se pulsa una tecla que hace rotar el contenedor del vehículo de modo que deje caer los círculos que transporte dentro del sensor triangular de la plataforma flotante.

Descripción técnica

Este proyecto se basa en la implementación de objetos bidimensionales sencillos. Estos se utilizan para combinar formando modelos más complejos. Todo esto está organizado en una escena que crea todas estas entidades y las gestiona para que ejecuten su funcionalidad.

Entidades simples

Se utilizan tres entidades simples con las cuales se generan estructuras más complejas. Todas las entidades reciben como parámetro un b2World y un b2Body Type.

Circle

Entidad sencilla que representa una circunferencia. Recibe como parámetros la posición de su centro y el radio de la circunferencia.

Cube

Entidad sencilla que crea un cubo de dimensiones variables. Recibe como parámetros la posición de su centro y dos valores de ancho y alto.

Edge

Entidad sencilla que crea una línea sólida. Con estas entidades se construyen la mayoría de elementos del escenario. Recibe como parámetros las coordenadas de sus dos vértices.

Entidades Complejas

Estas entidades están formadas a partir de la combinación de otras. Se utilizan para generar los comportamientos más elaborados y reciben los parámetros necesarios para definir todos los elementos simples que las componen. Para realizar las uniones entre las entidades se utilizan Joints.

Platform

Esta entidad está compuesta por tres Edges. Se utiliza para elevar al coche hasta la parte final del enunciado. Dentro de esta entidad se encuentra la lógica pertinente para gestionar su movimiento automático en función de la posición del coche. Para que esta lógica funcione correctamente se debe obtener una referencia al coche. La unión entre los tres Edges se realiza con Revolution Joints con un motor de alto torque y velocidad 0. Estos recrean el comportamiento deseado.

Carretilla

Esta entidad está compuesta por tres Cubes. Se utiliza para varios elementos dentro del ejercicio:

- La cesta que almacena las bolas antes de entregarlas al coche.
- La parte móvil del coche donde se almacenan las bolas y se depositan al finalizar.

Coche

Esta entidad consta de un Cubo que hace de cuerpo principal, dos Circles que hacen de ruedas conectadas por dos Revolution Joints que actúan de motores, y una Carretilla conectada con otro Revolution Joint para poder moverla.

Scene

En este proyecto la escena acapara dos funcionalidades. La primera es la propia función de albergar los elementos que la componen y los sistemas necesarios para que funcionen. La segunda es la de kernel, pues también orquesta el origen en el que estos sistemas se ejecutan y posee el core loop de ejecución. Los sistemas presentes por orden de ejecución son:

- Read Input
- Update
- Render

Diario de desarrollo

Este proyecto es el segundo realizado en este entorno y con este lenguaje. La mayor familiaridad con las librerías ha hecho que su desarrollo sea mucho más fluido. Ha habido decisiones mejorables pero inevitables debido a la falta de tiempo. Un ejemplo de esto es la no implementación del motor creado para la entrega anterior. Pues tras un primer intento se pudo apreciar que el tiempo que llevaría sería más elevado que empezar desde cero recreando algunos sistemas de forma simplificada.

Joints

El mayor obstáculo encontrado durante la realización del ejercicio fue el uso de los joints. Por puro desconocimiento se intentaron usar Distance Joints para la conexión de objetos obteniendo muy malos resultados. El desarrollo fue mucho más fluido al usar Revolution Joints y entender mejor el funcionamiento de estos así como sus diferentes tipos, usos y configuraciones

Reflexiones

Este proyecto ha sido bastante gratificante de realizar a pesar del poco tiempo del que se disponía para hacerlo por causas externas a la asignatura. El desarrollo ha sido bastante exitoso y el resultado muy vistoso, pues se trata de un pequeño juego creado enteramente con código C++.

Posibles mejoras

A nivel de mejoras ha habido funcionalidades que no han sido posibles implementarlas por falta de tiempo. Algunas de estas pueden ser:

- Utilizar un motor.
- Implementación del respawn del vehículo en caso de caída, volcado o petición del usuario.
- Implementación de las partículas de final de partida cuando se finaliza el ejercicio.

También podría haberse hecho un mejor trabajo en cuanto a los comentarios del código.

Conocimientos adquiridos

La realización de esta práctica ha permitido afianzar aún más los conocimientos del lenguaje C++, así como el uso de la librería box2D.

Este segundo punto ha sido sin duda el más relevante ya que tras esta práctica podría realizarse todo tipo de ejercicios similares o más complejos con el uso de esta misma librería.