

Escuela Superior de Informática

Universidad de Castilla-La Mancha

CURSO DE EXPERTO EN DESARROLLO DE VIDEOJUEGOS

TRABAJO FIN DE CURSO

*ACCEL-EXCEEDED*

JULIO 2015



Escuela Superior de Informática

Universidad de Castilla-La Mancha

CURSO DE EXPERTO EN DESARROLLO DE VIDEOJUEGOS

TRABAJO FIN DE CURSO

*ACCEL-EXCEEDED*

*Fernández Duran, Juan Carlos*

*Martínez Heras, Iván*

JULIO 2015



CURSO DE EXPERTO EN DESARROLLO DE VIDEOJUEGOS

Calificación Trabajo Fin de Curso

CONVOCATORIA: Julio 2015

TÍTULO DEL PROYECTO: *ACCEL-EXCEEDED*

AUTORES (ORDEN ALFABÉTICO): Fernández Duran, Juan Carlos

*Martínez Heras, Iván*

TRIBUNAL:

|  |  |
| --- | --- |
| Presidente: |  |
| Vocal: |  |
| Secretario: |  |

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE VOCAL SECRETARIO

Fdo: Fdo: Fdo:

Índice:

1. Introducción.
2. Resumen.
3. Objetivos.
4. Arquitectura del juego.
5. Análisis de costes.
6. Manual de usuario.
7. Conclusiones.
8. Propuestas y trabajo futuro.
9. Bibliografía.

1. - Introducción.

Con este proyecto se pretende realizar una demo para un juego en 3D totalmente funcional. Para dicho juego se pondrán en uso de los conocimientos adquiridos a lo largo del curso.

En este proyecto hemos usado el motor gráfico Ogre3D para manejar las escenas y todos los objetos tridimensionales, Bullet como motor de físicas para controlar las colisiones y la herramienta CEGUI para diseñar los menús del juego.

2. - Resumen.

Accel-Exceeded está concebido como un juego de plataformas a contrarreloj en el que los usuarios compiten por ocupar los mejores puestos en el ranking al modo de los juegos árcade tradicionales.

Tomando como referencia elementos de otros juegos comerciales como Portal, Mirrors Edge o Prince of Persia, Accel-Exceeded ofrece un nivel que requiere del uso de toda la destreza del usuario, además de saber utilizar en el momento oportuno las habilidades del personaje (doble salto, aceleración, retroceder en el tiempo) para ser superado en el menor tiempo posible.

3. - Objetivos.

RF1- El juego se desarrollará en Primera Persona, se debe implementar y es remarcablemente importante que esta sea fluida ya que es el principal mecanismo del juego.

RF2- Se implementará los controles clásicos de movimiento de todo FPS, deslizamiento lateral y avance frontal y marcha atrás (WASD), además, se implementará el doble salto.

RF3- Se implementará una habilidad secundaria que complemente el juego, en este caso, la marcha atrás en el tiempo, de forma limitada.

RF4- Se implementará un sistema de Records

RNF1- El juego se podrá ejecutar en Linux

RNF2- El juego se renderizará y ejecutará como mínimo a 30 FPS estables, preferiblemente 60

4. - Arquitectura del juego.

El juego se apoya principalmente en varios módulos, para el apartado gráfico, y su renderizado, se utiliza OGRE, se simulan físicas con Bullet utilizando como nexo entre el juego y el motor de físicas el wrapper OgreBullet, se utiliza OIS para la gestión de la entrada y salida además de otros usos, o CEGUI para el menú.

El juego se divide en un diagrama de estados ya estudiado en el curso, en el que tenemos IntroState donde tenemos todo lo referido a CEGUI, es decir, el Menú.

Para el uso de varios punteros se han tenido en cuenta las eficiencias del nuevo estándar C++11 y se han utilizado smart pointers, por ejemplo para el SceneNode del jugador o algunos gestores propios.

Un objetivo que se ha hecho hincapié durante este curso 2014/2015 es sobre el uso apropiado de las estructuras de datos para situaciones apropiadas, en este caso, la funcionalidad a implementar era retroceder en el tiempo, estudiemos detenidamente este “poder”

Periódicamente, de registrará la posición del jugador como un OGRE::Vector3 para poder utilizarlo posteriormente, siempre lo insertaremos al final de la cola, y si el tamaño de la estructura de datos rebasa un límite, comenzaremos a eliminar los datos del principio de la cola ya que son los que quedan obsoletos debido al tiempo transcurrido, y de esta forma limitamos al jugador para no poder utilizar este poder de forma tan prolongada.

Una vez estemos retrocediendo en el tiempo tan solo necesitaremos consultar el último dato, y una vez aproximemos al jugador lo suficiente a ese Vector3D, eliminaremos este último dato y si se continúa la vuelta atrás en el tiempo, en el siguiente frame realizaremos dicha traslación para el nuevo último dato, de esta forma damos el efecto de volver atrás en el tiempo en lo relativo a la posición del jugador.

La estructura de datos Vector no es apropiada para este propósito, ya que si  se desborda el tamaño y eliminamos al principio del vector, la estuctura de datos, de forma interna, se remodelará en la memoria a cada intervalo y eso penalizará gravemente nuestro rendimiento

La estructura de datos deque sin embargo, es ideal, ya que es constante la inserción, eliminación y acceso de datos tanto al principio como al final de la estructura de datos. Su principal desventaja es la eliminación o inserción en una posición arbitraria del deque, sin embargo no nos interesa realizar dichas modificaciones en nuestro juego.

Cabe destacar que la Primera Persona en Accel Exceeded se ha implementado manualmente, es decir, prescindimos del clásico cubo de colisión que simula las físicas del jugador, ya que Bullet daba severos problemas al interactuar con este cubo debido a que las fuerzas de fricción de la base del cubo con el suelo le daba una aspereza al control que no deseábamos para nuestro juego.

En su lugar, se ha implementado mediante Raycast las colisiones, en el que modificamos la traslación del jugador en función de la dirección de la cámara, y una velocidad definida por nosotras así como el movimiento lateral realizando giros de 90 grados con los vectores de comprobación de colisión.

También se ha implementado de forma manual la gravedad, sin embargo, lo más costoso de detectar y corregir ha sido la subida progresiva del jugador a medida que subía las escaleras, para ello se ha tenido en cuenta una distancia “anormal” del suelo respecto al jugador y se aplica una traslación ascendente para corregir dicha anomalía.

El sistema de records se gestiona al igual que otros juegos nuestros anteriores mediante un fichero de texto externo que es comprobado al iniciar al juego, de el se extraen los 3 tiempos menores y se muestran en orden

Para el desarrollo del juego se ha utilizado Vim como entorno de desarrollo para la parte de la programación, y de forma complementaria, Gedit, Tmux y Makefile para mayor comodidad del desarrollo.

5. - Análisis de costes.

Juan Carlos Fernández Durán:

Tareas:

->Gestión de Input.

->Implementación del movimiento en Primera Persona.

->Implementación de la habilidad de vuelta atrás en el tiempo.

->Implementación de sistema de records solo a nivel lógico (sin mostrarlo en el menú, registrarlos, y recuperar los 3 mejores).

->Implementación de condiciones de Victoria.

Iván Martínez Heras

Tareas:

->Diseño e implementación del nivel jugable.

->Diseño e implementación del menú del juego.

->Implementación de la música.

->Diseño e implementación del personaje y sus animaciones (descartado).

->Implementación de luces.

6. - Manual de usuario.

**W ->** Adelante.

**S ->** Atrás.

**A ->** Izquierda.

**D ->** Derecha.

**Espacio ->** Salto.

**Espacio-Espacio ->** Doble salto.

**B ->** Retroceder en el tiempo.

7. - Conclusiones.

A pesar de la sencillez del concepto, creemos que es un juego con grandes posibilidades y al que se le puede sacar mucho partido con algo de trabajo añadido. El desarrollo de este proyecto nos ha servido para profundizar mucho más en los conceptos estudiados durante el curso así como a descubrir malos hábitos acarreados de antes.

8. - Propuestas y trabajo futuro.

Como propuesta para ampliar el juego en un futuro está el diseñar e implementar una inteligencia artificial pensada para ser usada en modos para un jugador, como pudiese ser el modo carrera o el modo persecución.

Otra propuesta interesante que nos gustaría añadir seria un modo multijugador, ya que el juego da pie a ello. Podría implementarse a pantalla partida o incluso para jugar en red.

9. - Bibliografía.

http://en.cppreference.com/w/cpp/container/deque

http://en.cppreference.com/w/cpp/container/vector

http://static.cegui.org.uk/

http://www.ogre3d.org/docs/api/1.9/

http://wiki.blender.org/index.php/ES/Main\_Page