

Capítulo 2: Videojuegos: Una pasión y grandes desafíos

El desarrollo de videojuegos resulta ser un proceso demandante, ya que siempre se requiere estar actualizado tanto en conocimientos teóricos como en hardware. En este capítulo, se explicarán los principales problemas con los que se encuentra un desarrollador de videojuegos, algunos de los retos que conlleva el desarrollo del producto, su complejidad en términos técnicos y su diferenciación con respecto a cualquier otro tipo de aplicación informática.

2.1 Complejidad y retos en el desarrollo de videojuegos

Así como en el desarrollo de cualquier otro tipo de software, existen retos para poder llevar a cabo el proyecto del desarrollo de un videojuego, los desarrolladores deben dominar el lenguaje de programación usado, conocer las ventajas y desventajas de dicho lenguaje así como saber lidiar con los problemas típicos que se presentan en el desarrollo de software, tales como corrupción de memoria, errores en la inicialización de variables, definición de variables, pérdida de memoria, etc. Aunado a esto, el desarrollador de videojuegos se enfrenta a retos adicionales, los cuales son descritos por Wayne Ariola en su artículo “*The Challenges of Developing Games & Other High-Resolution Graphics*” de la siguiente manera:

2.1.1 No hay lugar para errores

Debido a los altos estándares en la industria de los videojuegos, no es permitido lanzar un juego con *bugs*. El software con fallos que sea lanzado puede tener un gran impacto en la imagen del estudio desarrollador y generar pérdidas en las ganancias estipuladas, ya que los jugadores difícilmente olvidan un *bug* o error en el juego que acaban de comprar. Por otra parte, los errores en el software

pueden producir al grupo desarrollador una gran carga financiera. Con el concentrado ciclo de ventas, que es común en las organizaciones encargadas en desarrollar videojuegos, cualquier imprevisto de *debugging*⁹ y reestructuración puede causar retrasos que tengan repercusiones económicas en la promoción, publicidad, envíos, etc.

2.1.2 Plazos ajustados

Las organizaciones enfocadas al desarrollo de videojuegos deben cumplir ciertos plazos de entrega que, por lo general, se ubican en una ventana de oportunidad de ventas. El ejemplo más claro es el mes de diciembre por Navidad. Por ejemplo, si un juego pierde la oportunidad de ser lanzado en esta época, la organización posiblemente no cuente con la oportunidad de recuperar la inversión hecha en el producto. Por otra parte, cuando una organización pierde la ventana de oportunidad a la cual el producto estaba enfocado, sus competidores aprovecharán esta oportunidad para sobrepasarlos con sus propios juegos.

El consumidor en este tipo de industria, por lo general, en su deseo por obtener lo último y lo mejor, es “distráido” del título que esperaba y fue retrasado y, como consecuencia, compra un producto similar.

En conclusión, debido a que los retrasos tienen consecuencias muy riesgosas en la industria, los calendarios de lanzamiento suelen ser muy agresivos y existe poca tolerancia a desplazarlos.

2.1.3 Código escrito por desarrolladores inexpertos o cansados

Las condiciones típicas en las que se desarrollan los videojuegos, proveen el ambiente perfecto para la introducción de errores en el código. En primera instancia, muchos desarrolladores de videojuegos son jóvenes y no cuentan con la

⁹ Debugging, proceso de resolver bugs o errores en el código de un programa

experiencia adecuada para reconocer y evadir errores comunes en la programación. En segunda instancia, debido a que hay típicamente muchos desarrolladores no experimentados trabajando en un juego, es común encontrar que el código está escrito en diferentes estilos. Al suceder esto, los otros desarrolladores del equipo malinterpretan la función del código, teniendo como consecuencia la implementación de errores basados en una mala interpretación. Por último, los desarrolladores de videojuegos generalmente trabajan por muchas horas y bajo presión, para poder cumplir con los plazos de entrega establecidos por la organización. Como consecuencia, hasta los programadores más experimentados, olvidan lo que han aprendido anteriormente y llegan a introducir errores en el código. En otro tipo de industrias existen personas dedicadas a revisar que el código, en todo el equipo, esté escrito de manera correcta y uniforme. Sin embargo, por lo dicho anteriormente sobre los calendarios de lanzamiento agresivos, es imposible contar con este tipo de apoyo en la industria de los videojuegos.

2.1.4 Múltiples niveles de evaluación

Cada plataforma de juegos, como *Xbox*, *PlayStation*, *Steam*¹⁰, etc. cuenta con una organización externa que se dedica a validar los juegos que son enviados para su publicación. Los juegos son evaluados desde la calidad en el código hasta la jugabilidad que plantean. Si existen problemas durante el proceso de validación, los desarrolladores deben arreglarlos en una fase tardía del desarrollo, cuando resulta más costoso, difícil y tardado. Un método para identificar este tipo de problemas en las primeras fases de desarrollo, debe de ser implementado para ahorrar trabajo y costos extras.

¹⁰ Steam, una plataforma de distribución de juegos para PC en línea.

2.1.5 El proceso de desarrollo no suele ser tan repetible como debería de ser

El desarrollo de videojuegos es, sin lugar a dudas, un proceso creativo. Sin embargo, los desarrolladores de videojuegos generalmente se resisten a implementar estándares y procesos repetibles, debido al miedo que les genera que este tipo de métodos limiten su creatividad. Irónicamente, este tipo de estándares y procesos, son los que pueden ayudar a asegurar que los frutos de la creatividad del desarrollador y su trabajo, sean pagados cuando el juego sea validado y se ponga en venta en el mercado a tiempo.

2.1.6 Se debe de interactuar con librerías gráficas

Los desarrolladores de videojuegos deben interactuar con diferentes librerías gráficas, para poder maximizar los efectos gráficos y así alcanzar el realismo que los consumidores demandan. En el desarrollo tradicional de aplicaciones, el paso final en el desarrollo es encargarse de errores en el código como regresar valores incorrectos, corrupción y pérdida de memoria. En el desarrollo de videojuegos, asegurarse que no ocurran este tipo de problemas es sólo el inicio.

Librerías gráficas, como *OpenGL*¹¹ o *DirectX*¹², cuentan con el soporte de gráficos realistas y de alta definición que esperan los consumidores. Debido a que los procesadores por sí solos no pueden manejar este tipo de gráficos, mucha de esta carga se ha trasladado a que la procesen las tarjetas gráficas. Estas librerías son las que se encargan de establecer el vínculo mediante el cual la aplicación comunica las instrucciones para generar el *rendering* en las tarjetas gráficas.

¹¹ OpenGL, es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.

¹² DirectX, es una colección de API desarrolladas para facilitar las complejas tareas relacionadas con multimedia, especialmente programación de juegos y vídeo, en la plataforma Microsoft Windows.

Si el código no interactúa de manera correcta con las librerías gráficas, los gráficos generados no serán los que se diseñaron en un principio y muchos de los movimientos podrían parecer irreales.

En la gran mayoría de las aplicaciones informáticas, los gráficos suelen ser al aspecto que mejora la aplicación, sin embargo, en los videojuegos, los gráficos suelen ser la aplicación, al menos desde la perspectiva del consumidor. Los errores en gráficos son un punto crítico en el desarrollo de videojuegos, los desarrolladores deben aprender a dominar tanto las tecnologías más usadas como las emergentes.

2.2 Retos técnicos

Los videojuegos se desarrollan en una pantalla que refleja de forma gráfica la gran cantidad de procesos que se llevan a cabo en una plataforma. La plataforma puede ser una computadora, un teléfono celular o una consola especializada en juegos, como lo es *PlayStation*, *Xbox* o *Nintendo*.

Su objetivo es entretener al usuario. Los jugadores deben actuar en diversas situaciones virtuales, controlando a un personaje o cualquier otro elemento del entorno; mismo que ha sido creado virtualmente para cumplir los objetivos que son impuestos como reglas del juego. El videojuego se puede jugar solo, contra una inteligencia artificial, o con más jugadores, ya sea en la misma consola o por medio de una red, como por ejemplo, el Internet. La forma de interacción del jugador con el videojuego es por medio de dispositivos externos, tales como el teclado de una computadora y mouse, o un mando de juego. Actualmente existen nuevas tecnologías que permiten la detección de movimiento del cuerpo jugador para dar un enfoque totalmente nuevo a la interacción.

Normalmente, se distribuyen por medio de cartuchos o discos y se pueden conseguir fácilmente, tanto en supermercados como, en tiendas especializadas. Gracias al fácil acceso a Internet con el que contamos en la actualidad, se creó un

modelo nuevo de ventas mediante la red; *Apple App Store*, *Google Play Store*, *Steam* y *Origin*¹³, entre otros, han logrado destacar como plataformas para la distribución de videojuegos en línea.

2.2.1 Diferencias con otro tipo de software

La programación de videojuegos, si bien a primera vista parece ser algo difícil y complejo, dista mucho de ser irrealizable. El programador que decide realizar su primer juego, siempre se enfrenta a la desventaja de desconocer cómo estructurar su código. En la universidad y en los cursos que se imparten para desarrollar la habilidad de programar, generalmente se enseña enfocado al desarrollo de aplicaciones informáticas, dejando a los videojuegos como algo que se tenía que aprender de manera autodidacta. Afortunadamente, gracias al acceso a la información que nos provee el Internet, resulta más sencillo el aprendizaje enfocado al desarrollo de este tipo de software.

Comparado con las aplicaciones informáticas, no existen demasiadas diferencias estructurales entre ellas y un videojuego. Su diferencia primordial, es que la aplicación suele realizar una sola tarea simultánea, es decir, la que corresponde a la opción activa en ese momento, mientras que el videojuego **procesa muchas tareas a la vez**. Cada personaje, enemigo, obstáculo u objeto que aparece en pantalla, e inclusive los que no han llegado a plasmarse en ese momento, tienen que ser procesados de manera simultánea. Esto plantea una dificultad adicional al programador, ya que se deben prever todas las posibilidades que pueden resultar de la interacción de todos los procesos que se están ejecutando.

Para crear un videojuego es necesario tener una **visión global**. Se debe de ser capaz de ver el conjunto de procesos que representan cada uno de los aspectos del juego y como interactúan entre sí. Otro aspecto fundamental en el

¹³ Medios de distribución digital de contenidos. En orden, sus respectivos dueños son: Apple Inc., Google Inc., Valve Inc., Electronic Arts Inc.

desarrollo de videojuegos es la necesidad de monitorear el tiempo; requiere un elevado control del tiempo transcurrido desde que se inició el juego o desde que ocurrió un determinado evento en el tiempo de ejecución. Independientemente de la velocidad con la que permite el hardware ejecutar el juego, es necesario que el tiempo sea medido de una forma exacta.

Al igual que una aplicación informática, un juego puede ser dividido en varias partes. Las tareas a realizar para poder completar la fase técnica, pueden dividirse en bloques según la parte del juego que gestionen. Debe de existir una serie de procesos que tengan como tarea actualizar las posiciones de los objetos que se mueven en pantalla, tales como el personaje o los enemigos; existirán una o varias funciones que se encarguen de mostrar las imágenes del juego en pantalla; otra parte del juego se debe de encargar de regir las leyes físicas del juego, es decir, detectar las interacciones con el medio, la fuerza de gravedad y las colisiones por nombrar algunos ejemplos, entre los distintos elementos del juego. Los aspectos generales del juego, tales como son el tiempo, puntaje, generación y eliminación de objetos y enemigos, también deben ser controladas por otra parte del programa.

Los videojuegos no son aplicaciones que se ejecuten normalmente dentro de una ventana, generalmente ocupan toda la pantalla. Bajo esta perspectiva, es normal que cuando se inicie el juego, todas las demás aplicaciones que se mostraban en pantalla, en el caso de que ejecutemos el juego en una computadora, sean sustituidas por una pantalla de presentación, acompañada generalmente de una serie de menús del juego.

Otra característica que distingue a los videojuegos es la forma en la que se maneja el ciclo de mensajes. Toda aplicación para computadora tiene un módulo que se dedica a **gestionar los mensajes**. Por ejemplo, al momento de pulsar una tecla o de mover el mouse sobre cierta región de la pantalla, se realiza cierta acción. Normalmente, si no sucede ningún evento dentro de la ventana de la aplicación, se supone que el usuario no la está usando en ese momento; ésta pasa a un estado de espera, en el cual disminuye la cantidad de recursos del

sistema que utiliza hasta que se genere un nuevo evento dentro de su respectiva ventana. De esta forma, se ahorran y se aprovechan al máximo los recursos del sistema. Sin embargo, un videojuego no suele tener estados de espera, aunque no se reciba ningún evento por parte del jugador, los objetos que se encuentran en pantalla deben permanecer activos y realizando las tareas que les fueron programadas.

En conclusión, los videojuegos son aplicaciones que **consumen una gran cantidad de recursos** del sistema, se ejecutan de manera independiente y exclusiva en la plataforma de hardware, requieren de una planeación detallada de los eventos que ocurrirán en el juego donde el tiempo juega un papel fundamental en el desarrollo de éste.

2.2.2 Especificaciones técnicas del hardware

Como lo hemos venido mencionando anteriormente, el videojuego se ejecuta sobre una plataforma de hardware, cada una de éstas posee diferentes capacidades técnicas; como la cantidad de memoria volátil, el espacio de almacenamiento, la velocidad del procesador, por mencionar algunas. Es necesario saber, desde antes de comenzar a desarrollar el videojuego, en qué tipo de plataforma se va a ejecutar para, de esta forma, tomar en cuenta las limitaciones y herramientas con las que se puede contar.

2.2.3 Poder de procesamiento (Procesador principal, GPU)

El poder de procesamiento es una parte fundamental en el proceso del desarrollo de un videojuego. Si no se cuenta con la suficiente velocidad y capacidad para procesar cada uno de los eventos que ocurren, la diversión que genera jugar un videojuego se convierte en frustración, ya que el juego puede

volverse lento debido a una pérdida en el *frame rate*¹⁴, absurdo o, en el peor de los casos, injugable.

2.2.4 Memoria volátil (RAM, VRAM)

Debido a lo complejos y realistas que se han vuelto los videojuegos en los últimos años, es necesario contar con una gran cantidad de memoria volátil para poder almacenar y tener un rápido acceso a las diferentes variables, texturas y modelos que forman parte de los muchos otros datos que se cargan y/o se generan en tiempo real. En el caso de no contar con la suficiente cantidad de memoria volátil, el juego se verá reducido principalmente en sus capacidades visuales.

2.2.5 Memoria no volátil (Disco duro, CD, DVD, etc.)

Este tipo de memoria se utiliza para almacenar el juego, si estamos hablando de un disco duro, en éste es donde se almacenan los archivos necesarios para la reproducción del juego como podrían ser los videos, la música, las texturas, etc. La gran mayoría de estos datos, son trasladados a la memoria volátil al momento de ejecutar el juego, esto se hace debido a que se cuenta con un mejor y más rápido acceso a éste tipo de memoria que a un disco duro.

En el caso de los CD, DVD o cualquier otro medio de almacenamiento óptico, por lo general se utilizan para distribuir el juego y luego se procede a instalar los recursos más necesitados en un disco duro.

¹⁴ Frame rate: Es el número de cuadros por segundo que se despliega en pantalla. Por lo general, en videojuegos, se utiliza un estándar de 60 ó 30 cuadros por segundo.

2.3 Adaptación del proyecto a nuevas tecnologías

2.3.1 Simulación cercana a la realidad

Actualmente, los desarrolladores tratan de implementar en casi todos los géneros de los videojuegos elementos que se asemejen a la realidad. Ya no solamente consiste en realizar la trayectoria recta de una bala en un disparo, se toman en cuenta factores como el viento, la distancia, el culatazo del arma y el tipo de bala que se utiliza. De igual manera para un juego de carreras de autos, no basta con sólo imitar a la perfección cada uno de los modelos de las distintas marcas que aparecen en el juego, se debe de tomar en cuenta el tipo de terreno por el cual el auto circula, las llantas con las que cuenta, el coeficiente de fricción de éstas, el tipo de clima que está en la pista, la altitud con respecto a nivel del mar, etc.



Figura 2.1: Gran Turismo 5: The real driving simulator

En la figura 2.1, se puede apreciar el gran grado de complejidad que maneja el simulador *Gran Turismo 5*¹⁵. Recrea perfectamente el interior de la cabina de un automóvil y los movimientos del piloto; todo esto adicional a la simulación de la pista, el terreno y las leyes de física aplicadas al automóvil.

2.3.2 Fluidos

Resulta sumamente complicado realizar una simulación de como se comportaría un fluido, tal como el agua. Los videojuegos tratan de imitar las características de los fluidos en el mundo real para transportarlos a un mundo digital, esto obviamente requiere de una gran cantidad de procesamiento para poder hacerlo lo más realista posible. Los desarrolladores de videojuegos, se encuentran experimentando con sistemas de partículas, en los que grupos de partículas responden a eventos mientras cumplen ciertas normas.

En la Figura 2.2, se puede observar el comportamiento del agua en un videojuego. Se nota el detalle en los reflejos de las luces y los objetos sobre el fluido, así como también su permeabilidad.



Figura 2.2: Ejemplo del comportamiento de fluidos

¹⁵ Gran Turismo 5, un simulador desarrollado por Polyphony Digital y publicado por Sony Computer Entertainment. Ha vendido, hasta finales del 2010, más de 6,370, 000 copias.

2.3.3 Fuego

Programar el comportamiento del fuego es muy similar a trabajar con fluidos, su diferencia radica en que, el fuego, se mueve más violentamente y cuenta con la capacidad de consumir a otros objetos hasta sus cenizas, generando gases derivados de la combustión.

En la Figura 2.3, podemos ver un ejemplo de cómo se realiza la simulación del fuego, en *Team Fortress 2*¹⁶.



Figura 2.3: Ejemplo del comportamiento del fuego

2.3.4 Expresión humana

Desde hace mucho tiempo, se han analizado los rasgos que caracterizan al ser humano, muchos artistas y científicos han tratado de observar un patrón en el comportamiento humano. Hoy en día, los desarrolladores de videojuegos intentan crear personajes más humanos, dotándolos de características que psicólogos han investigado.

¹⁶ Team Fortress 2, un juego exitoso desarrollado por Valve Inc.

Para generar reglas que satisfagan la gran cantidad de combinaciones que existen en la expresión y emoción de un ser humano, desarrolladores del estudio *Valve* se han basado en la taxonomía de un total de 60 acciones fáciles básicas, definidas por los psicólogos Paul Ekman y Wallace V. Friesen. En la siguiente imagen, podemos observar diferentes expresiones faciales del comandante Shephard, personaje protagónico en la serie *Mass Effect*¹⁷.



Figura 2.4: Ejemplo de rostros generados en “Mass Effect”, desarrollado por Bioware

2.3.5 Inteligencia artificial

Como consecuencia de la exigencia de crear juegos más reales, la inteligencia artificial *IA* se ha vuelto más compleja. Los consumidores quieren tener retos al enfrentar enemigos así como también contar con inteligentes aliados.

Dotar a los personajes secundarios de la capacidad para poder tomar decisiones como en la vida real, consume una gran parte de los recursos del

¹⁷ *Mass Effect*, una trilogía desarrollada por Bioware, ha creado un universo tan rico que se han publicado novelas y cómics para complementar su historia; inclusive se le ha llegado a comparar con *Star Wars* de George Lucas o *The Lord Of The Rings* de Tolkien.

procesador; se deben emplear teorías lógicas de alto nivel, árboles de decisión, autómatas de estado finito, tal y como los usan en el diseño de robots. Es importante mencionar, que no queremos que la IA se comporte de manera perfecta, ya que volvería el juego sumamente difícil y tedioso para el jugador, lo que se pretende es simular el comportamiento humano, que dista mucho de ser perfecto.



Figura 2.5: Un claro ejemplo de IA, sería el que ejemplifican los juegos de deportes, tal como FIFA 2013.

2.3.6 Luz y sombras

Es preciso recordar que, a diferencia de las películas animadas por computadora, un videojuego desarrolla los gráficos en tiempo real. El desarrollador no puede dejar trabajando la computadora por días como lo haría un animador, la responsabilidad del desarrollador de videojuegos es crear efectos de iluminación asemejados a la realidad, tomando en cuenta el poder del procesador. Entre más elementos existan en el juego, más compleja y pesada se vuelve la tarea de

generar la iluminación; si el jugador se encuentra en una jungla, por ejemplo, se debe de procesar la entrada de luz al ambiente por cada árbol, hoja y planta que se encuentre en la vegetación, todo esto aunado al procesamiento a la generación de sombras de cada uno de los elementos que forman parte de la jungla, tales como el personaje, los enemigos y la misma vegetación.



Figura 2.6: Gráficos en tiempo real generados por Unity3D Versión 4.0

Existen empresas que se dedican a facilitar este tipo de tareas al desarrollador, creando motores de juego y licenciándolos. Un ejemplo lo podemos ver en la figura 2.7 con la compañía *Crytek* con su última versión de *CryEngine*, éste ha logrado generar un excelente manejo de iluminación en tiempo real gracias a su módulo *Polybump 2* que se encarga de reducir una superficie compleja a menos polígonos.

2.3.7 Física en los materiales

Si se pretende crear una simulación casi real en un mundo virtual, es necesario clasificar los materiales que forman parte de este mundo para así poder

describir el comportamiento de cada uno. Las leyes físicas rigen de la misma manera a un ladrillo que a un trozo de algodón. Sin embargo, no generará las mismas consecuencias el tirar un trozo de algodón desde un décimo piso a que si fuera un ladrillo. Existen propiedades físicas tales como el módulo de Young, que permite evaluar la rigidez de un objeto, y el coeficiente de Poisson, que permite calcular el que tan flexible puede llegar a ser el objeto.



Figura 2.7: Simulación realizada en CryEngine 3



Figura 2.8: Ejemplo de explosión en Crysis 2, desarrollado por Crytek

2.3.8 Movimiento realista

Suele ser igual de complejo de simular que la expresión humana debido a la gran cantidad de músculos y articulaciones que posee el cuerpo humano. Si quisiéramos realizar este tipo de animaciones en tiempo real, animar a un solo personaje supondría un gran consumo de recursos del sistema. Para poder contar con una animación más fluida del cuerpo humano, se ha recurrido a técnicas de captura de movimiento. En la figura 2.9, logramos ver el rostro de una persona plagado de sensores que permiten la captura de los movimientos faciales. También, se aprecia que cuenta con un traje especializado y se encuentran cámaras alrededor del cuarto, mismas que permiten la precisa captura de todos sus movimientos.



Figura 2.9: Forma tradicional de realizar Motion Capture

2.3.9 Captura de movimiento

Los métodos usados anteriormente para capturar cada uno de los movimientos de un ser humano, involucraban que una persona se vistiera de un traje lleno de sensores y se moviera alrededor de un cuarto con cámaras. Este tipo de detección resultaba muy costoso y demandante de tiempo, ya que se debía de calibrar a la perfección cada uno de los sensores y de las cámaras en la habitación.

Existen nuevas tecnologías como *Organic's Motion's Stage*, que permiten que se capture el movimiento de una persona, con cualquier tipo de ropa y en cualquier tipo de cuarto. El sistema está basado en la alimentación de datos, provenientes de 10 a 14 cámaras, que triangulan la posición del cuerpo mediante cálculos realizados en una computadora; la Figura 2.10 nos da un ejemplo de la captura de movimientos.

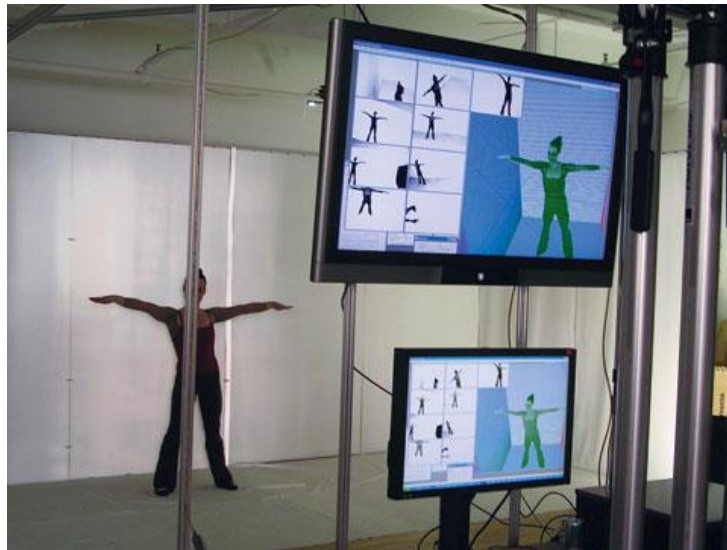


Figura 2.10: Organic's Motion's Stage nueva forma de capturar un movimiento realista.

2.4 Diversidad de formatos (3D – 2D)

Como describimos anteriormente, existen diferentes géneros en los videojuegos, de acuerdo al género en el cual se basara el juego que se pretende realizar, es el formato que debe de adoptar. Por ejemplo, si se pretende realizar un juego de plataformas, existen dos tipos de entornos, basados en una vista en 2D o permitir al jugador desplazarse libremente por los tres ejes dimensionales. De la misma manera, si se pretende hacer un juego de disparos en primera persona, lo lógico sería realizar un ambiente en tres dimensiones.



Figura 2.11: Megaman X4, un juego de acción-plataformas en dos dimensiones.

Cada uno de los géneros de videojuegos cuenta con ciertas características que los definen, se recomienda que el equipo de desarrollo investigue sobre el género que pretenden implementar y decida qué características son las óptimas y realizables para su juego, así como también si la plataforma de hardware escogida es la óptima. Es necesario mencionar que, de acuerdo a las características de la plataforma en donde se va a ejecutar nuestro juego, es como se podrán

implementar y realizar las mecánicas de éste. Es decir, un juego realizado para PC enfocado a un navegador, podría no resultar divertido de jugar en un Smartphone. Es necesario tomar en cuenta las características con las que contará el producto final, para así poder escoger qué plataforma es la que más se adecúa a él.



Figura 2.12: Halo 4, un first person shooter para Xbox 360

2.5 Retos administrativos

El desarrollo de software implica diversos retos administrativos que se derivan de la entrega en tiempo y forma de un proyecto de calidad para un determinado cliente. Para lograr tal objetivo, es necesario realizar una correcta planeación sobre del desarrollo del proyecto, en la cual se deben incluir los recursos con los que se cuentan y, de esta forma, determinar si dicha meta es viable. Dentro de estos recursos, es indispensable incluir al factor humano, pues es éste quien agrega otro nivel de complejidad al proyecto. No es fácil coordinar los esfuerzos de todo un equipo heterogéneo de personas hacia un fin común; organizar el trabajo en equipo, transmitir la correcta visión del proyecto a cada uno

de los integrantes, controlar avances y mantener un nivel adecuado de calidad resultan ser actividades demandantes. Finalmente, no se debe dejar de lado, como en cualquier trabajo en equipo, la existencia de conflictos y diferencias dentro del equipo y el desarrollo del proyecto; es necesaria la resolución rápida y eficiente de éstos para lograr el cumplimiento eficaz del objetivo propuesto. Es así como el desarrollo de software se percibe como una tarea de suma complejidad y verdaderamente demandante.

Las actividades anteriormente mencionadas son realizadas por un administrador de proyectos, quien define el tipo de administración que se utilizará dependiendo del tipo de proyecto, la naturaleza del equipo y el ambiente bajo el cual se trabajará. Los tipos de administración son mejor conocidos como metodologías de desarrollo y están pensadas para resolver diversos problemas administrativos que, se sabe, son recurrentes en proyectos de una determinada índole.

Hemos observado que el desarrollo de videojuegos contiene gran variedad de retos a nivel técnico como administrativo y por lo tanto se debe buscar elementos que ayudan a atacar esta gran variedad de retos que se han presentado. Por tanto, en el siguiente capítulo se explicará la manera en que las metodologías de desarrollo de software ayudan a enfrentar estos retos.