

Diseño e Implementación de un Juego para Dispositivos con Android: SpeedyRoad

Renso Alessandro Sucari Velasquez, Eyyvind Osco Cuba

Universidad Católica San Pablo

Interacción Humano Computador

I. RESUMEN

En este trabajo se presenta el diseño e implementación de un videojuego para dispositivos con sistema operativo Android. El objetivo principal de este proyecto consiste en aprender a desarrollar este tipo de aplicaciones, resolver las contingencias comunes inherentes a las limitaciones hardware que puede ofrecer un dispositivo móvil y entender el ciclo de vida de un proyecto como éste. Para el desarrollo de este trabajo se ha utilizado el entorno de desarrollo Unity (Motor de desarrollo de videojuegos multiplataforma) programando en C. Así cualquier persona pueda acceder a ella sin mucho costo y sin muchas especificaciones. El servidor de juegos dedicado extensible y funcional se produjo como resultado del proceso de desarrollo, marcando el proyecto como un éxito y dando una base para futuras implementaciones.

II. INTRODUCCIÓN

Un videojuego es una aplicación interactiva orientada al entretenimiento en el que una o más personas interactúan, por medio de un controlador, con un dispositivo que permite simular experiencias en las que intervienen los jugadores a través de un dispositivo visual. Los videojuegos necesitan de un determinado dispositivo electrónico, conocido como plataforma, para poder ser utilizados. Puede ser un ordenador, una máquina arcade, una videoconsola o un dispositivo portátil (un teléfono móvil, por ejemplo). Al dispositivo de entrada usado para manipular un videojuego se lo conoce como controlador de videojuego o mando, y varía dependiendo de la plataforma. Por ejemplo, un controlador podría únicamente consistir de un botón y un joystick, mientras otro podría presentar una docena de botones y una o más palancas (gamepad).

Los primeros juegos solían hacer uso de un teclado para llevar a cabo la interacción, o bien requerían que el usuario utilizara un joystick con un botón como mínimo. Muchos juegos de ordenador modernos permiten o exigen que el usuario utilice un teclado y un ratón de forma simultánea. Entre los controladores más típicos están los gamepads, joysticks, kinect, teclados, ratones y pantallas táctiles. Por lo general, los videojuegos hacen uso de otras maneras, aparte de la imagen, de proveer la interactividad e información al jugador. Un claro ejemplo de esto es el audio, usándose dispositivos de reproducción de sonido, tales como altavoces y auriculares; o la vibración, como en los casos de algunos mandos de videoconsolas más modernas y terminales

móviles. El desarrollo de videojuegos desde finales del año 2000 ha sufrido una importante evolución hacia nuevas plataformas de ejecución. Los videojuegos hasta la fecha, prácticamente estaban limitados a ordenadores y videoconsolas, abriendose un nuevo mundo para los desarrolladores: los teléfonos móviles de última generación (smartphones) y tabletas con pantallas táctiles.

Estos dispositivos tienen unas capacidades gráficas que les permiten ejecutar juegos que son técnicamente comparables a los de mediados de los años 90 para ordenadores de sobremesa. Las tiendas de aplicaciones (App Store para iOS y GooglePlay para Android) permiten a los usuarios descargar e instalar fácilmente estos juegos. Los sistemas de pago integrados permiten a un cliente comprar sin problemas juegos completos o hacer compras en el juego (sistema de micro-pagos). Unity [1] es actualmente el motor de juego más utilizado para el desarrollo de videojuegos para plataformas móviles. Es compatible con casi todas las plataformas en las que los juegos se ejecutan en la actualidad: Ordenadores de sobremesa, portátiles, consolas, navegadores web y smartphones. Cada uno de estos dispositivos utiliza la función multijugador a su manera, pero un factor común es que utilizan servidores para permitir este nivel de interacción.



Fig. 1. Plataforma 3D

III. MOTIVACIÓN

Muchas de las personas que de alguna manera han tenido contacto con alguna de las disciplinas necesarias para el desarrollo de videojuegos se han preguntado alguna vez: ¿Cómo se hace un videojuego?, ¿qué tengo que saber?, ¿por dónde empiezo? Y un largo etcétera. Muchos han intentado

hacerlo sin resultados concluyentes; otros en cambio han podido crear o participar en sus pequeños juegos con un poco de esfuerzo, pero aun así siempre quedan preguntas no resueltas y mucho por aprender. Existe gran cantidad de libros y artículos relacionados con el desarrollo de videojuegos.

Es un mundo muy amplio en el que cada individuo tiene que tomar su propio camino para iniciarse en él con un objetivo (por pequeño que sea) marcado, e ir aprendiendo todo lo posible para conseguir la meta deseada. Para estar al día con la tecnología que se aplica en el mundo de los videojuegos, hay que ser amplio de miras y renovarse constantemente para tener todos los elementos necesarios: Es fundamental conocer la evolución de las herramientas de desarrollo para utilizarlas en nuestro beneficio. Para comenzar en el desarrollo de videojuegos es preferible hacerlo con proyectos pequeños, con el fin de ser capaces de cumplir las metas que nos fijamos y nunca pensar en proyectos extremadamente grandes que se nos escapen de las manos, ya que lo más probable es que nunca los terminemos, lo cual nos llevará a la frustración. Es común comenzar con remakes de juegos como Pong, Tetris, Pacman, etcétera. Es muy buen ejercicio desarrollar algún juego conocido con modificaciones. Nos hará aprender infinidad de soluciones a problemas que ni llegábamos a imaginar, lo que hará enriquecernos con muchos conocimientos que podremos aplicar en futuros desarrollos.

Cada individuo tiene que tener claro cuál es el fin cuando dedica tiempo y esfuerzo en el desarrollo de un videojuego. En el caso de tener como objetivo un desarrollo profesional en el mundo de los videojuegos, es preciso estudiar y entender todos los procesos que se ven involucrados, desde los llevados en una pequeña empresa de desarrollo hasta los que se llevan a cabo en los grandes estudios de la industria.

Abarcando nuestro proyecto, la estructura principal contiene los requisitos del servidor y lo que debe ser capaz de hacer. A esto le sigue la introducción de la teoría detrás de las redes y los servidores de juegos en general. También se explican las herramientas que proporciona para crear este servidor. La teoría es seguida por la práctica, donde la implementación es discutida y respaldada por la teoría y los requisitos manejados previamente. El detalle de la implementación es el capítulo principal y el enfoque de este proyecto. Una vez que se maneja la implementación, es hora de evaluar el proceso de desarrollo e implementación y deducir lo que se podría haber hecho de manera diferente y lo que aún queda por hacer.

IV. OBJETIVOS

El objetivo principal de este proyecto es conocer y aprender todas las fases del desarrollo de un videojuego de plataformas para dispositivos móviles. Todo ello, a través de la tecnología de desarrollo Unity, ampliamente utilizada para el desarrollo de juegos. En base a este objetivo principal, se proponen los siguientes objetivos:

- Estudiar el funcionamiento del entorno de desarrollo



Fig. 2. Speedy Road Multiplayer

Unity, enfocado particularmente a plataformas móviles(multiplayer).

- Aprender a utilizar el lenguaje de programación C#, orientado al desarrollo de videojuegos sobre todo en la interacción con el usuario, para que esta fluya de la mejor manera.
- Diseñar y documentar, con cierto nivel de abstracción, el conjunto de eventos y situaciones que se darán en el juego a desarrollar.
- Dotar al juego de un nivel mínimo de jugabilidad. Se tiene que poder jugar y que “aporte diversión” al usuario.
- Desarrollar una aplicación que permita añadir nuevas funcionalidades con facilidad. Debe estar modularizado y debe haber una documentación amplia acerca del funcionamiento de cada uno de los métodos y propiedades.

V. TRABAJOS RELACIONADOS

Se abordó el tema sin conocimiento previo de las implementaciones de multijugador de Unity y, por lo tanto, todas las epifanías son una gran combinación de teoría, conocimiento previo por lo que fue una enorme cantidad de investigaciones, ensayos, errores y experimentación. El trabajo previo relacionado con las partidas multijugador en el juego se realizó con la mente en el multijugador de host a cliente y el producto viable mínimo. Esto proporcionó una gran cantidad de trabajo y estructura para el servidor, pero también introdujo varias partes que requirieron reimplementación y rediseño de programación y funcionalidad. El trabajo presentado en este documento abre muchas funcionalidades clave del núcleo del servidor del juego y la lógica de coincidencia del juego.

VI. PROPUESTA

VI-A. Descripción

Nuestra aplicación multiplayer(servidor y clientes), estos conectados a una red, brindada por un tercero. Se basa en la competencia de 2 autos(el primero deja obstáculos, el segundo tiene que evitarlos), el segundo auto tiene la posibilidad de hacer todas las maniobras posibles para evitar el daño causado por estos obstáculos.



Fig. 3. Menu Inicio



Fig. 5. Dejando Cono

VI-B. Tecnologías Usadas

- Android
- Unity3D

VI-C. Eventos de Interacción

- Selección: El usuario podrá escoger una de las opciones brindadas por el menu inicio, ya sea para iniciar al juego, configuración o salir de este.
- Navegación: El usuario podrá navegar libremente en el auto haciendo avanzar el auto.
- Manipulación: El usuario mediante el movimiento del giroscopio podrá brindarle dirección del auto (izquierda,derecha).
- Sonido: Tenemos dos clases de sonido ambiental y sonido propio del carro, lo cual aumenta la experiencia real.

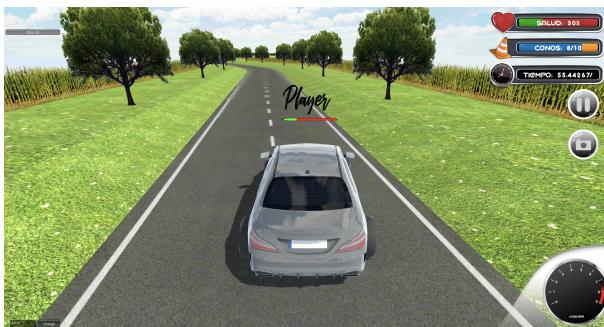


Fig. 4. Choque

VI-D. Acciones Especiales

Un jugador deja un obstáculo(cono) mediante dos toques en la pantalla, este cono tiene un numero límite de creaciones, son creados dentro un tiempo determinado, desaparecen, ocasionan daño al otro jugador.

VI-E. Salud

Cada cono ocasionará 10 puntos de daño al jugador que lo toque, siendo la vida máxima 100 puntos, cuando la vida llegue a 0 puntos el jugador sera enviado al inicio de la pista

VI-F. Conexión de red

El servidor del juego es una entidad con el propósito de ser el lugar de reunión para dos instancias del juego del cliente. El ancho de banda disponible no debe ser saturado o abusado de otra manera con sincronizaciones de estado constante. En su lugar, las sincronizaciones de estado deben integrarse en los eventos del juego que el servidor de juegos percibe como importantes para que los clientes puedan replicar y sincronizar sus estados.

El servidor del juego debe ser capaz de manejar las conexiones de clientes recibidas con la capacidad de aceptar a los clientes en el juego, almacenar la información de los clientes durante la duración de la partida y desconectar a los clientes también. El lado del cliente del juego utilizado por el jugador no debe bloquearse ni lanzar una excepción cuando la conexión se desconecta del servidor del juego.

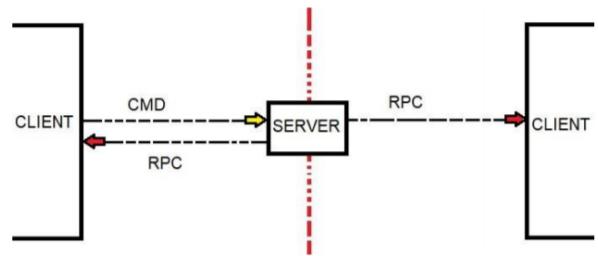


Fig. 6. Interacción Servidor-Clientes

VII. PRUEBAS Y RESULTADOS

Se describe el conjunto de experimentos que han sido realizados sobre la aplicación desarrollada en este proyecto. Estos experimentos han sido definidos para analizar los requisitos de la aplicación establecidos inicialmente en este documento. Dichos experimentos consisten en un conjunto de pruebas que cada usuario tiene que realizar, para posteriormente responder a varias preguntas relacionadas con el resultado de las pruebas. Los objetivos principales de la experimentación son:

- Comprobar que el flujo de la aplicación no contiene fallos.
- Observar el grado de fiabilidad de la aplicación respecto al diseño de la misma.

- Conocer las opiniones de los usuarios acerca de la jugabilidad y usabilidad.
- Recoger sugerencias de los usuarios para que puedan ser aplicadas en trabajos futuros.

Se hizo pruebas con varios dispositivos Android y usuarios que interactuaron con la aplicación y nos dieron los siguientes resultados.

- El problema parecía ser de unos doscientos milisegundos a unos pocos segundos. En relación con el retraso en la espera de la conexión adecuada del cliente, fue trivial, pero en cuanto a la espera del usuario fue bastante significativo.
- Algunos usuarios percibieron que algunos de los controles no funcionan correctamente.
- Usuarios con nivel de experiencia bajo, les pareció complicado la aplicación
- Los 6 tipos de cámaras les pareció un exceso ya que no dependen mucho de estas, ya que están más centradas en evitar los conos.
- Debido a las texturas, los usuarios vieron que se volvía lento al cambiar de escena o al regresar al menu inicio.
- En cuanto a los controles, se tuvo que asistir al principio a un par de usuarios, especialmente con el doble salto. Tras unas breves indicaciones, los usuarios se adaptaron bien a las mecánicas de la aplicación.
- A la pregunta de si ha gustado el juego, todos los usuarios lo han encontrado muy divertido, lo que cumple con uno de los pilares fundamentales de los videojuegos: Ha de ser entretenido.

Se debe tener en cuenta que la sincronización continua y las llamadas de enganche de SyncVar de múltiples representantes de la clase aumentan considerablemente el tráfico de la red. Para aliviar estos problemas, varias variables SyncVar se descartaron de sus atributos y se modificaron para que funcionen sin ellas durante el desarrollo. Es práctico sincronizar las variables relacionadas con el juego percibido del cliente y las variables como los tiempos internos, si no se comunican al jugador, es mejor dejarlas fuera de las variables sincronizadas.

VIII. CONCLUSIONES

El objetivo principal desde el principio del proyecto consistía en conocer y aprender todas las fases de desarrollo de un videojuego de plataformas para dispositivos móviles(multiplayer). Este objetivo se ha conseguido satisfactoriamente, ya que al tener tras finalizar este proyecto un juego de plataformas para móviles con un flujo de juego y usabilidad coherentes, implica que se ha entendido cada una de las fases de desarrollo de este tipo de software. No obstante desarrollar videojuegos de cualquier tipo que puedan ser utilizados en diferentes plataformas es muy complicado. A lo largo del desarrollo de este proyecto se ha visto que el desarrollo de un videojuego implica una serie de dificultades y restricciones que son complicadas de definir a priori.

Es vital realizar un estudio previo de las tecnologías que se quieren usar, tanto las tecnologías para el desarrollo como las

plataformas objetivo, pues las conclusiones de este estudio definirán las “reglas del juego” para poder desarrollar la aplicación satisfactoriamente. El diseño de la arquitectura del software y la previsión en el desarrollo a la larga ahorra una cantidad de tiempo muy valiosa.

se constató que el juego a priori era algo difícil, hecho que es complicado de ver por el desarrollador por estar demasiado acostumbrado a las mecánicas del juego. Tener una buena estructura para el futuro es importante, ya que tener un juego jugable en el público hace que sea más difícil implementar nuevas funciones y funcionalidades. Los aprendizajes realizados durante el desarrollo han dado mucha estructura para futuras modificaciones, que son principalmente optimizaciones y mayor escalabilidad en las funcionalidades. Las implementaciones futuras del servidor de juegos para otros proyectos son probablemente más fáciles cuando se realizan en conjunto con este proyecto debido a la cantidad de conocimiento y las descripciones de las funcionalidades proporcionadas.

VIII-A. Trabajos Futuros

- Incremento de niveles y mundos temáticos
- Puntuación
- Mejoras y personalización
- Enemigos
- Inteligencia de los enemigos

REFERENCES

- [1] Unity Documentation [Available]: <https://unity3d.com/>
- [2] Android Studio [Available]: <https://developer.android.com/guide/?hl=es-419>
- [3] PEA, Roy D. User centered system design: New perspectives on human-computer interaction. 1986.
- [4] Ignatchenko, S. 2017. Development and Deployment of Multiplayer Online Games, Volume 3. 2nd beta. Leanpub.
- [5] Unity Documentation 2017. ClientRpcAttribute. Available at: <https://docs.unity3d.com/ScriptReference/Networking.ClientRpcAttribute.html>
- [6] Unity Documentation 2017. CommandAttribute. Available at: <https://docs.unity3d.com/ScriptReference/Networking.CommandAttribute.html>