

INFORME

1. Contexto problemático

El equipo de Epic Games, que es una de las empresas encargadas de desarrollar uno de los videojuegos que en los últimos tiempos ha tenido un gran reconocimiento mundial, como lo es FORTNITE. Requiere de la modificación de algunas funciones que no han sido del total agrado de los amantes a este juego, implementando diferentes estructuras de datos para lograr el objetivo planteado por la empresa.

2. Desarrollo de la solución

Para resolver el caso anterior se eligió el Método de la Ingeniería del libro "Introduction to Engineering" de Paul Wright. Cada uno de los pasos del Método de la Ingeniería se presentan a continuación especificando la descripción y el procedimiento de cada uno de estos pasos.

2.1 Identificación del problema

De acuerdo al enunciado se pudieron detectar los siguientes problemas:

1. Realizar un sistema que permita que los usuarios sean rankeados de acuerdo al nivel que han alcanzado dentro del juego y de esta manera tener combates con usuarios que posean un nivel en el ranking similar.
2. Comprobar que todos los usuarios que se encuentran dentro de una partida tengan una latencia muy cercana, asegurando de esta manera la igualdad de condiciones entre los participantes de dicha partida.
3. Implementar en el juego el modo "Plataforma", para que de esta forma se pueda limitar la entrada de participantes a un combate, dependiendo de la plataforma que estén utilizando.
4. Desarrollar el modo de juego que se usará en SAN VALENTÍN. El cual consiste en que el jugador solo podrá usar la última arma que ha recogido a lo largo de la partida.

2.2 Recopilación de la información.

Para una debida solución de nuestro problema inicial, requerimos recopilar información que nos ayude a esclarecer la manera mas optima para poder resolverlo. de esta manera entenderemos el problema a fondo con todas las implicaciones que este contenga. Además de generar diferentes ideas que nos permitirá tomar el mejor camino para llevarlo a cabo. para esto, clasificaremos la información en dos secciones. la primera, será recopilación del juego fornite

como tal, para entender un poco su funcionamiento. y la segunda sección será de las estructuras de datos que nos ayudarán a elaborar esta labor.

Recopilación de la información del juego.

Fornite es un juego del género battle royale, el cual consiste básicamente en un juego que combina los elementos de un videojuego de supervivencia con la jugabilidad de un último jugador en pie. Los videojuegos tipo *battle royale* desafían a un grupo de jugadores, comenzando con un equipamiento mínimo, a que busquen armas y eliminen a otros oponentes, mientras evitan quedar fuera de un "área segura" (el campo de juego se hace más pequeño), siendo el ganador el último jugador en pie.

Armas.

Fornite cuenta con una variedad de armas de todo tipo, escopetas, francotiradores, pistolas, fusiles, subfusiles, granadas, misiles, etc. Estas armas se pueden encontrar de cinco formas diferentes.

- Distribuidas aleatoriamente en el mapa
- En cofres
- En las conocidas "llamas".
- En los drops que caen cada cierto tiempo en el mapa.
- En las máquinas dispensadoras, que se encuentran en diferentes lugares del mapa.

Para la elaboración de este trabajo, sólo se tendrán en cuenta las armas que se encuentran distribuidas aleatoriamente en el mapa.

Gameplay.

Jugar al fornite es relativamente sencillo. El juego consiste en agrupar 100 jugadores (a veces, menos) en un mapa, en este caso es una isla. Consiste en conseguir desde el minuto uno en armas, materiales y consumibles (salud y/o escudo) para que el jugador vaya sobreviviendo. Cada cierto tiempo, la tormenta se acerca, es decir que el mapa se vuelve más pequeño, logrando así que jugadores puedan encontrarse con más facilidad para combatir. En estos combates el único objetivo es conseguir que la salud del oponente llegue a 0. Una variedad que tiene fornite a los otros battle royale, es el modo de construcción, pues en medio del combate cada jugador puede generar estructuras con sus materiales para protegerse y hacer un combate más interesante. Teniendo todo lo anterior en cuenta, el ganador del juego será quien quede en pie, es decir, quien quede vivo de último. Es decir, para ganar

una partida de fornite no se necesita matar a todos, solo basta con matar al último jugador en pie.

Matchmaking.

Este tema ha sido uno de los más controversiales del juego, pues tanto como jugadores profesionales, como los aficionados no están de acuerdo con esto. No existe como tal un proceso de selección de partida que permita una distribución más justa. Por ejemplo, no es justo que se empareje a un jugador con nivel 5 con otro de nivel 100, o que un jugador que haya ganado 10 partidas, se empareje con otro con 600 partidas ganadas. En este problema los desarrolladores del juego han trabajado y prometido que mejoraran su matchmaking, pero hasta ahora las quejas siguen siendo las mismas.

Plataformas.

Fornite es un juego que a lo largo de su desarrollo ha ido aumentando las plataformas en las que se encuentra. por lo pronto el juego se encuentra disponible en Xbox, Play 4, Pc, Android, ios, Nintendo Switch.

Recopilación de información para la solución del problema.

Latencia.

La latencia en cambio es el tiempo exacto que tarda en transmitirse un paquete dentro de la red, el tiempo que tardas en recibir un paquete del servidor. Lo que mide es la inmediatez de la conexión, en caso de que tengamos una latencia de 50 milisegundos significa que ese es el tiempo que tardan en llegar desde, por ejemplo, los servidores de Google hasta nuestro ordenador.

Volviendo a la analogía de antes, podemos tener una carretera de tres carriles por la que circulan cientos de coches, pero eso no significa que estemos viendo el tráfico en tiempo real. Pueden pasar unos milisegundos desde que un coche pasa hasta que nosotros lo vemos pasar, y eso es exactamente la latencia.

Para medir la latencia se utiliza el *ping*, que se mide en milisegundos (ms) el tiempo que tardan en comunicarse tu conexión local con un equipo remoto en la red IP. Por eso, cuando vas a realizar un test de velocidad lo primero que hace la web que lo va a realizar es medir el Ping, y luego ya pasa a darte las velocidades de bajada y subida de datos.

Pilas.

Una pila (stack en inglés) es una lista ordinal o estructura de datos en la que el modo de acceso a sus elementos es de tipo LIFO (del inglés Last In First Out, último en entrar, primero en salir) que permite almacenar y recuperar datos. Se aplica en multitud de ocasiones en informática debido a su simplicidad y ordenación implícita en la propia estructura. Representación gráfica de una pila

Para el manejo de los datos se cuenta con dos operaciones básicas: apilar (push), que coloca un objeto en la pila, y su operación inversa, retirar (o desapilar, pop), que retira el último elemento apilado.

En cada momento sólo se tiene acceso a la parte superior de la pila, es decir, al último objeto apilado (denominado TOS, Top of Stack en inglés). La operación retirar permite la obtención de este elemento, que es retirado de la pila permitiendo el acceso al siguiente (apilado con anterioridad), que pasa a ser el nuevo TOS.

Por analogía con objetos cotidianos, una operación apilar equivaldría a colocar un plato sobre una pila de platos, y una operación retirar a retirarlo.

Colas.

Cola (informática). Una cola es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción push se realiza por un extremo y la operación de extracción pop por el otro. También se le llama estructura FIFO (del inglés First In First Out), debido a que el primer elemento en entrar será también el primero en salir.

Las colas se utilizan en sistemas informáticos, transportes y operaciones de investigación (entre otros), donde los objetos, personas o eventos son tomados como datos que se almacenan y se guardan mediante colas para su posterior procesamiento. Este tipo de estructura de datos abstracta se implementa en lenguajes orientados a objetos mediante clases, en forma de listas enlazadas.

HashTable

Las tablas hash son estructuras de datos que se utilizan para almacenar un número elevado de datos sobre los que se necesitan operaciones de búsqueda e inserción muy eficientes. Una tabla hash almacena un conjunto de pares "(clave, valor)". La clave es única para cada elemento de la tabla y es el dato que se utiliza para buscar un determinado valor.

Un diccionario es un ejemplo de estructura que se puede implementar mediante una tabla hash. Para cada par, la clave es la palabra a buscar, y el valor contiene su significado. El uso de esta estructura de datos es tan común en el desarrollo de aplicaciones que algunos lenguajes las incluyen como tipos básicos.

tomado de:

http://www.it.uc3m.es/abel/as/MMC/M2/HashTable_es.html

<https://www.epicgames.com/fortnite/forums>

<https://www.xataka.com/basics/que-son-el-ping-y-la-latencia-y-por-que-no-solo-importa-la-velocidad-en-tu-conexion>

[https://www.ecured.cu/Cola_\(Estructura_de_datos\)](https://www.ecured.cu/Cola_(Estructura_de_datos))

2.3 Búsqueda de soluciones creativas.

Para este paso todas las ideas daban una solución al problema de una o cierta manera. Para poder realizar un mejor desarrollo, decidimos dividir las soluciones en tres.

1. solución al problema del ranking de los jugadores.
2. solución al problema del ping de los jugadores.
3. solución al problema de partida por plataforma.
4. solución al problema del modo san valentín (limitación del inventario).

Para la solución de nuestros cuatros problemas, tenemos alternativas que se basan en almacenar datos, pues en los cuatro ítems se desarrollara el problema con estas estructuras. De esta manera, cada problema será solucionado con una estructura de datos en particular.

alter

2.3.1 solución al problema de ranking de los jugadores(matchmaking)

- **Alternativa 1:** Realizar un método de ordenamiento de los anteriormente vistos de tal forma de que los jugadores nos queden organizados de manera descendente de acuerdo a su nivel en el ranking, y de esta manera va a ser mucho más sencillo encontrar a varios usuarios con un nivel igual o muy cercano.

- **Alternativa 2:** Realizar un método de búsqueda, el cual reciba el nivel del usuario que se está registrando, y de esta manera encontrar los jugadores que tienen ese mismo nivel o que varíe en una cantidad mínima.
- **Alternativa 3:** Puesto que todos los usuarios registrados deben aparecer con el nivel al que pertenecen, permitir que el jugador que está ingresando elija todos aquellos que tengan el mismo nivel o muy similar.
- **Alternativa 4:** Hacer uso de una cola para organizar de una mejor manera los usuarios y de esta forma hacer un matchmaking óptimo que permita que esta función se desarrolle como debe ser.

2.3.2 Solución al problema de partida por plataforma

- **Alternativa 1:** Cuando el usuario vaya a ingresar al juego permitirle que escoja en qué plataforma desea jugar, ya sea PC, Play 4, Dispositivo Android, Dispositivo iOS, XBOX ONE. Y de esta manera ser separados por plataforma, donde los tipos de plataformas se encuentran en una lista.
- **Alternativa 2:** Que el usuario tenga un atributo plataforma y realizar un método de búsqueda donde se encuentren a los usuarios que estén registrados con el mismo tipo de plataforma.

2.3.3 Solución al problema del modo San Valentín (limitación del inventario)

Es un modo de juego donde solamente el usuario puede hacer uso de la última arma que haya encontrado y en el cual se puede cambiar exclusivamente si se le acaban las municiones o en caso de encontrar otra arma. Se debe tener en cuenta que la única arma que no va a desaparecer es el hacha, pues esta no tiene municiones.

- **Alternativa 1:** Hacer uso de una pila, en la cual se puede utilizar el método PUSH para agregar o apilar la última arma que el jugador haya encontrado en el juego y de esta misma manera el método POP que realiza la acción inversa que es desapilar o retirar un objeto, en este caso el arma, de la pila.
- **Alternativa 2:** Hacer uso de una cola, al igual que la pila, usa los métodos PUSH para agregar un objeto a la cola y POP para eliminar un objeto de la misma. La cola se caracteriza porque el primer objeto en entrar será el primero en salir.
- **Alternativa 3:** Utilizar un arreglo, en el cual uno de sus objetos será la hacha que se le entrega a todo jugador que ingrese al juego y la cual no desaparecerá, por lo explicado anteriormente, y otro objeto que tendrá será el arma que encuentre, que en caso tal de que se le acaben sus municiones o encuentre otra esta será eliminada y sustituida por la nueva arma encontrada.

- **Alternativa 4:** Utilizar un ArrayList, el cual se vaya llenando con todas las armas que vaya recogiendo en el camino, pero que el usuario solamente tenga la posibilidad de usar la última que haya encontrado, mientras que el resto se encuentren inhabilitadas.

2.4 Transición de las ideas a los diseños preliminares

2.4.1 Solución al problema del ranking de los jugadores

- **Alternativa 1:** El ordenar los usuarios de acuerdo a los atributos anteriormente mencionados es una solución que cumple con el problema. Pero la dificultad que puede ocurrir es la complejidad temporal que llevará esto, pues tendremos casi 2 millones de usuarios, que al momento de ordenar tomará un tiempo no eficiente para el programa.
- **Alternativa 2:** tenemos que tener en cuenta que la búsqueda será para un jugador en específico que tenga un ping igual o similar. lo cual nos entregaría un solo jugador. si deseamos crear una sala con 100 jugadores, el proceso se tendría que repetir 99 veces lo cual no sería óptimo.
- **Alternativa 3:** Esta solución sería fácil de implementar, pero si la hacemos estaremos dándole la responsabilidad al usuario, el cual no debe preocuparse por este problema. por esta razón, no es considerado como una alternativa final.
- **Alternativa 4:** Al hacer uso de una cola, podríamos almacenar en esta estructura los jugadores que tengan similar ping, lo cual permitiría que uno a uno vayan llegando a la cola de “ping similar” y así se logren clasificar en una misma partida, lo cual sería un matchmaking óptimo.

2.4.2 Solución al problema del modo plataformas

- **Alternativa 1:** Pedirle esta información al usuario no es tan difícil, y al darnos esta información podríamos tener a todos los jugadores clasificados de acuerdo a su plataforma en una lista. Lo cual permitirá su clasificación en el MatchMaking
- **Alternativa 2:** De nuevo el mismo problema, la búsqueda se tendría que hacer muchas veces, lo cual haría que la complejidad temporal aumente, así entonces nuestro programa no sería óptimo.

2.4.2 Solución al problema del modo San Valentín

- **Alternativa 1:** Al hacer uso de la pila nos facilita el ingreso de las armas, pues dado a que solo se puede usar la última recogida, esto nos indicaría que sería la que esté en el tope de la pila, así que en términos de estructuras de datos es la mas optima
- **Alternativa 2:** Con la cola, el proceso de usar la última arma que se recoge, es decir, usar el primero en entrar sería un problema, pues para salir, el arma tendría que entrar a la cola y pasar por todo el conjunto hasta llegar a la salida, lo cual dificulta el requerimiento del problema.
- **Alternativa 3 y 4:** las dos alternativas claramente nos ayudará con la solución del problema, pues acceder al arma que deseamos, agregarla, o eliminarla son procesos que son fáciles de realizar, pero al compararlas con la alternativa 1, no son tan eficientes al momento de actuar.

2.5 Evaluación y Selección de la Mejor Solución:

De acuerdo a las observaciones que le hicimos a cada alternativa encontramos al final la aternativa que cumplira con el objetivo de manera óptima.

- Para el problema del matchmaking usaremos una cola.
- Para el problema de las plataformas usaremos una lista
- Para el problema del modo de juego san valentín se usará una pila.

2.6 Preparación de Informes y Especificaciones:

Especificación del Problema de MatchMaking (en términos de entrada/salida)

- Problema: Originar una partida de Fornite donde el matchmaking sea óptimo, que el ping y destreza sea tomado en cuenta para la elección de los jugadores.
- Entradas: el conjunto de todos los jugadores que estén buscando partida.
- Salida: un conjunto de jugadores con atributos similares, es decir, un matchmaking óptimo.

Especificación del Problema de plataformas (en términos de entrada/salida)

- Problema: clasificar los jugadores y por ende las partidas de manera óptima, que el ping y destreza sea tomado en cuenta para la elección de los jugadores.
- Entradas: tipo de plataforma que usa cada jugador.
- Salida: la clasificación de los jugadores por plataforma para el matchmaking

Especificación del Problema de MatchMaking (en términos de entrada/salida)

- Problema: Originar una partida de Fornite donde el matchmaking sea óptimo, que el ping y destreza sea tomado en cuenta para la elección de los jugadores.
- Entradas: el conjunto de todos los jugadores que estén buscando partida.
- Salida: un conjunto de jugadores con atributos similares, es decir, un matchmaking óptimo.

consideraciones:

- el matchmaking se realizará teniendo en cuenta la destreza del jugador y su ping.
- las plataformas permitidas en este ejercicio, serán pc, nintendo switch, xbox, psp.
- el pico (axe) será considerado como un objeto fijo en el inventario del jugador.
- para el modo de juego, solo se tendrán en cuenta las armas en el inventario. las curaciones, trampas y demás objetos propios del juego no serán tomados en cuenta.

2.7 Implementación del Diseño:

- los requerimientos funcionales serán entregados en un archivo adicional
- el código del programa será entregado en lenguaje Java, usado mediante Java eclipse.

