

Método de la Ingeniería

1) Identificación del Problema:

a) Necesidades:

- i) Proponer una forma eficiente para acceder a los bloques del sistema
- ii) Implementar los features que los jugadores de Minecraft quieren para poder realizar construcciones de manera mas comoda y rapida
- iii) Proponer una nueva interfaz de usuario que acople los nuevos cambios realizados y así atraer a nuevos jugadores

b) Definición del problema:

Minecraft necesita de unos cambios en el almacenamiento de su inventario debido a que el algoritmo usado es muy lento y poco eficiente, por lo que se necesita un algoritmo que ayude al juego a ser más eficiente. Además, gran parte de los jugadores de Minecraft piden mejorar la experiencia de juego de tal manera que la construcción pueda ser mas rapida y comoda para todos los jugadores, por lo que se requiere implementar dichas opciones que los jugadores quieren.

2) Recopilación de Información:

a) Conceptos:

- i) **Bloque:** Entidad principal de los objetos en el inventario, la característica principal de los objetos clasificados en este grupo es que pueden usarse para la construcción.
- ii) **Stack:** Término usado para identificar la máxima cantidad de bloques de un mismo tipo que se puede almacenar en una misma casilla.
- iii) **Modalidad o "Feature":** Es el servicio que presta una herramienta del sistema del juego para optimizar o mejorar la jugabilidad.
- iv) **Acceso rápido:** Es una prestación en la que una fila del inventario está presente en todo momento, ya sea que todo el inventario esté abierto o el jugador esté interactuando con el juego de cualquier forma, que permite la interacción directa de los objetos en esta con el mundo de minecraft sin necesidad de abrir el inventario.
- v) **Construcción:** Es el uso de bloques en la barra de acceso rápido mediante la colocación de estos en el mundo de la partida.

3) Búsqueda de Soluciones Creativas:

Se necesita mejorar tiempos ejecución y búsqueda entonces usaremos un tipo de estructura de datos que nos ayude en estos casos, omitiendo tipos de datos (Array, ArrayList)

Para acceder objetos de forma eficiente en el inventario podríamos usar alguna estructura de datos eficiente que nos ayude a solucionar el problema.

A) STACK:

Se busca el elemento que se obtuvo dentro de la pila para agruparlo en su conjunto que está en el inventario si existe ese conjunto y si no usar otro espacio de inventario para reunir más de ese elemento.

- B) HASH TABLE: Buscamos la llave a la que está asociado el objeto y agregamos el objeto a su grupo si no está lleno y si está lleno lo ponemos en otro slot del inventario

Para implementar la feature de acceso rápido podríamos usar las siguientes estructuras de datos:

- A) Queue: Creamos una cola como acceso rápido para el jugador donde va ingresando el objeto y el primer objeto que van entrando es el objeto que va usando.
- B) Stack: Creamos una pila para que el jugador pueda agrupar sus objetos y pueda buscar los elementos que requiera.
- C) Hash table: Agregaremos una hash table para agregar los objetos al acceso rápido y que al ingresar un objeto se busque por su clave.

Para implementar la feature de construcción debemos:

- A) Queue: Crearemos n colas para ingresar los objetos que necesita el usuario
- B) Stack: Creamos n pilas para ingresar los objetos que necesita el usuario
- C) Hash table: Creamos n hash tables para ingresar los objetos que necesita el usuario.

4) Transición de la Formulación de Ideas a los Diseños Preliminares

Para acceder objetos de forma eficiente en el inventario podríamos usar alguna estructura de datos eficiente que nos ayude a solucionar el problema.

- Stack: No es una opción viable ya que para usar Stack se debería crear un Stack por cada slot que existe en el inventario, por lo que no sería eficiente al guardar todo en memoria.

Para implementar el feature de acceso rápido se descartan las siguientes alternativas:

- Hash table: No es una opción viable este tipo de estructura de datos, ya que todos los elementos a guardar son de un mismo tipo, por lo que las keys todas serían iguales y su función hash enviará todo a la misma posición, y por esto no sería óptimo ya sería lo mismo que usar una estructura de datos más sencilla y probablemente más eficiente

Para implementar el feature de construcción se descartan las siguientes alternativas:

- Stack: No es viable porque para poder tener una lista por cada tipo de objeto, necesitaríamos crear n número de stacks, siendo n el número de tipos de objetos que se tengan en el inventario para poder así satisfacer todas las barras de acceso que pueda tener el usuario.
- Queue: No es viable porque para poder tener una lista por cada tipo de objeto, necesitaríamos crear n número de queues, siendo n el número de tipos de objetos que se tengan en el inventario para poder así satisfacer todas las barras de acceso que pueda tener el usuario.

5) Evaluación y Selección de la Mejor Solución

Definiremos los siguientes criterios, con base a una evaluación para definir la mejor solución:

Criterio A. Precisión de la solución. La alternativa entrega una solución:

[2] Exacta (se prefiere una solución exacta)

[1] Aproximada

Criterio B. Eficiencia.

Se prefiere una solución con mejor eficiencia que las otras consideradas. La eficiencia puede ser:

[4] Constante

[3] Mayor a constante

[2] Logarítmica

[1] Lineal

Criterio C. Completitud. Se prefiere una solución que encuentre todas las soluciones.

Cuántas soluciones entrega:

[3] Todas

[2] Más de una si las hay, aunque no todas

[1] Sólo una o ninguna

Criterio D. Facilidad en implementación algorítmica:

[2] Compatible con las operaciones aritméticas básicas de un equipo de cómputo moderno

[1] No compatible completamente con las operaciones aritméticas básicas de un equipo de cómputo moderno

Evaluación:

Evaluando los criterios anteriores en las alternativas que se mantienen, obtenemos la siguiente tabla:

Para acceder de forma eficiente los objetos del inventario

	Criterio A	Criterio B	Criterio C	Criterio D	Total
Stack	1	1	2	2	6
Hashtable	2	4	3	2	11

Para implementar el feature de acceso rápido

	Criterio A	Criterio B	Criterio C	Criterio D	Total
--	------------	------------	------------	------------	-------

Stack	2	4	3	2	11
Hashtable	1	1	2	2	6
Queue	2	4	3	2	11

Para implementar el feature de construcción:

	Criterio A	Criterio B	Criterio C	Criterio D	Total
Stack	1	1	2	2	6
Hashtable	2	4	3	2	11
Queue	1	1	2	2	6

Selección:

En la solución para hacer de forma eficiente la obtención de un objeto se opta por la hash table ya que en su evaluación obtuvo el más alto desempeño resolviendo el problema de manera eficaz.

Para el acceso rápido, dos de ellas obtuvieron el mismo desempeño en criterios similares ya que estas dos soluciones nos ayudan de manera precisa para la solución sin que una pierda calidad en su proceso

Para el feature de construcción: se opta por la hash table ya que en su evaluación obtuvo el más alto desempeño resolviendo el problema de manera precisa y eficaz.