

Inferno

By Gonzalo Vade and Tamara Sepúlveda

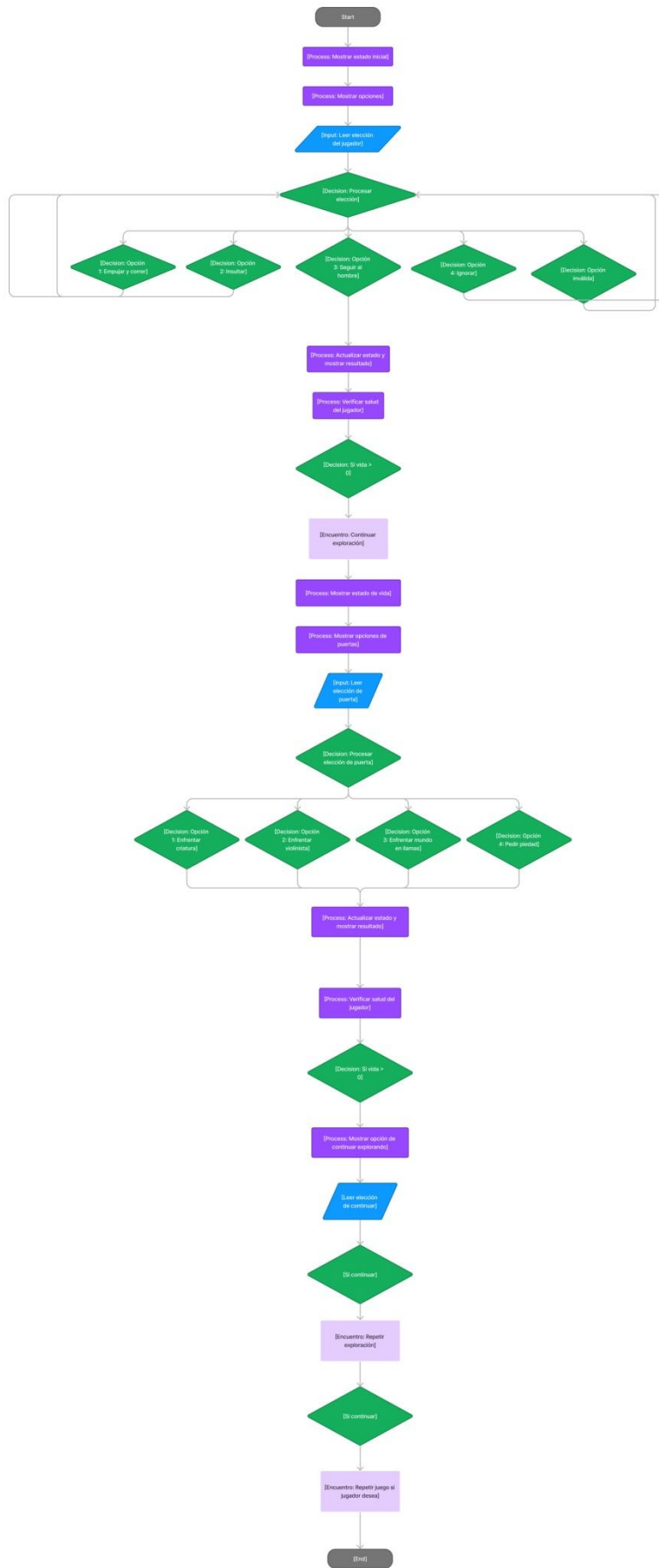
"Inferno" es un juego de aventuras y decisiones donde el jugador se encuentra atrapado en una mansión oscura y abandonada. El objetivo es explorar y tomar decisiones que afectarán el destino del protagonista. Durante la aventura, el jugador enfrenta encuentros inesperados y debe navegar a través de diversas situaciones para intentar escapar con vida.

Lógica Utilizada para Abordar el Juego El juego está programado en C++ y utiliza funciones y estructuras de control para manejar la narrativa y las interacciones con el jugador. Se utilizan funciones como `encounter` para representar encuentros cruciales y `continueExploration` para continuar la exploración según las decisiones del jugador. Además, se emplea la función `clearScreen` para limpiar la pantalla y proporcionar una interfaz más limpia y fácil de entender.

El juego gestiona la salud del jugador (`playerHealth`) como un recurso clave que se actualiza según las decisiones tomadas por el usuario. Las opciones presentadas al jugador están diseñadas para influir en la historia y determinar su éxito o fracaso en el juego. Explicación de Cómo el Programa Proporciona una Solución Efectiva Al iniciar el juego, se muestra al jugador su nivel actual de vida y se le presenta una serie de decisiones que afectarán el curso de la historia. Cada elección tiene consecuencias diferentes, desde perder vida hasta alterar el desarrollo de la trama.

El jugador puede elegir entre seguir a un extraño, explorar puertas misteriosas o ignorar las advertencias, cada una llevando a escenarios distintos que determinan el avance del juego. El juego continúa en un ciclo donde, tras cada encuentro y exploración, se da al jugador la opción de continuar jugando o finalizar la partida. Esta estructura de bucles y condicionales asegura una experiencia interactiva y dinámica, donde las acciones del jugador tienen un impacto directo en el resultado final.

Conclusiones "Inferno" ofrece una experiencia de juego envolvente y emocionante, donde las decisiones estratégicas y la gestión de recursos son clave para progresar. La combinación de narrativa interactiva y elementos de sorpresa asegura que cada partida sea única, fomentando múltiples jugadas para descubrir todos los posibles caminos y desenlaces. El código implementado proporciona una estructura sólida y eficiente para manejar la complejidad del juego, utilizando técnicas de programación como funciones, control de flujo y gestión de estado para crear una experiencia de usuario inmersiva y satisfactoria.



- **Encuentro** ahora se entiende como la fase general donde ocurren las interacciones principales del jugador dentro del juego, desde el inicio hasta la exploración y las decisiones tomadas en cada etapa.
 - **Start** y **End** marcan claramente el comienzo y el final del flujo del juego.
 - **Decision** se refiere a las opciones que el jugador puede elegir en cada punto de decisión.
 - **Process** incluye las acciones que realiza el juego para mostrar información al jugador o actualizar estados.
 - **Input** indica dónde el jugador proporciona una respuesta o toma una decisión.
-