Instituto Tecnológico de Costa Rica

Ingeniería en Computación

Lenguajes de Programación

Proyecto 4

Profesor: Ing. Allan Rodríguez Dávila

Estudiante: Carnet:

Jordano Escalante 2018161994

Segundo Semestre

Fecha: 26 de noviembre

Año:2023



**Tabla de contenidos**

Descripción del problema……………………………….………………………………….2

Manual de usuario……………………………………….………………………………….2

Instrucciones de compilación………………………………………………….…………...2

Instrucciones de ejecución…………………………………………………………………2

Instrucciones de uso…………………………………………………………………….…..3

Pruebas de funcionalidad…………………………………………………………….…….3

Diseño del programa……………………………………………………………..…….…...4

Librerías usadas…………………………………………………………………..…………4

Análisis de resultados………………………………………………………………..……..4

Bitácora………………………………………………………………………………..……...5

**Descripción del problema**

El proyecto es una aplicación web realizada por medio de express y JavaScript, la aplicación debe permitir al usuario crear una partida del juego snake y por medio de un servicio web permitir a otros usuarios unirse a la partida y empezar a jugar con las reglas del juego clásico snake.

**Manual de usuario**

Primeramente, sebe dirigirse a la dirección del repositorio público mediante el enlace <https://github.com/EscalanteWizard/snake.git>

Clone el repositorio en la ubicación que desee e ingrese a la carpeta snake que acaba de ser clonada en su equipo.

Ahora ubíquese dentro de la carpeta backEnd y en la consola ejecute el comando npx nodemon server.js, espere hasta ver el mensaje que indica que el servidor está siendo ejecutado en el puerto.. (puede aparecer el puerto número 3000,4545 o 4000).

Seguidamente ingrese a la carpeta frontEnd y en la consola ejecute el comando npx live-server index.html, en caso de que el administrador de permisos de windows le pregunte si desea que la aplicación realice cambios en su equipo seleccione la opción “aceptar”, eso desplegará el navegador que usted tenga seleccionado como principal y desplegará las opciones para crear una partida o unirse a una, seleccione la opción que desea y siga las instrucciones que el juego le indique.

**Instrucciones de compilación**

Este proyecto no necesita de compilación, lo que si es necesario es poner el puerto del backEnd en escucha antes de inicializar el frontEnd para asegurarse de que el puerto del socket bacnkEnd está escuchando antes de que el frontEnd haga alguna solicitud al mismo y no encuentre el servicio disponible.

Para poner el backEnd en escucha ubíquese en la carpeta server del proyecto y ejecute el comando npx nodemon server.js, si el proceso se completa con éxito deberá aparecer un mensaje que indique el número de puerto en el que se encuentra escuchando el servicio.

Una vez el servicio está en escucha ubíquese en la carpeta fronEnd del proyecto y ejecute el comando npx live-server index.html o en visualStudioCode presione clic derecho sobre index.html y seleccione a opción “ejecutar con live server”.

**Instrucciones de ejecución**

En la carpeta server, ejecute el comando npx nodemon server.js, una vez aparezca el mensaje que indica el puerto donde se está ejecutando el programa vaya a la carpeta frontEnd del proyecto y ejecute el comando npx live-server index.html y siga y cree su partida o bien únase a una existente ingresando el código de la misma en el espacio indicado.  
**Pruebas de funcionalidad**

**A screenshot of a game

Description automatically generated**

Do Ventana principal de inicio de juego, se despliega al momento de ejecutar el frontEnd.

A screenshot of a computer

Description automatically generated

Ventana de juego, la imagen muestra la versión single player, en la imagen se muestra el estado semilla donde se puede apreciar la serpiente del jugador representado por una línea blanca y la “comida” de la serpiente representada por un punto rojo.

**Diseño del programa**

En El programa debe contar con la comunicación entre dos aplicaciones distintas, frontEnd y backEnd, para ello se utilizó sockets para la comunicación.

Aparte de eso el programa utiliza listas y objetos de JavaScript para el manejo de salas de juego, jugadores, puntuaciones y el estado del tablero de juego.

Sin embargo no fue posible ahondar mayormente en la funcionalidad debido a problemas de errores y ambiente de desarrollo.

**Librerías usadas**

socket.io= para la comunicación entre servicios computacionales.

cors= validacion de servicios y credenciales

express= manejo de servidor web

**Análisis de resultados**

|  |  |
| --- | --- |
| Requerimiento | Completitud |
| Crear partida | 50% la partida se crea, pero no se despliega el estado inicial semilla ni se modifica el espacio de juego en función del numero de jugadores. |
| Unirse partida | 50% se despliega el tablero, pero no su estado inicial ni el tamaño del tablero en función del número de jugadores. |
| Ver ranking | 0% problemas de ambiente de ejecución |
| Despliegue de espacio de juego | 50% se ve el tablero, pero no se setean jugadores ni comida en el tablero |
|
|
| NickName de usuario | 0% problemas de ambiente y librerías, además poca experiencia con diseño web. |

**Bitácora**

La bitácora de trabajo puede ser consultada en el repositorio público de GitHub del siguiente enlace: https://github.com/EscalanteWizard/snake.git