

# M06UF1\_AA2 - Entrega Final:

## I l'estadi vibra amb la gran lluita!

### Documentación del Proyecto Pengti

### Documentación del Proyecto Pengti

Bugs

Explicación WebSockets:

Esquema conexiones Proyecto

- Conexión inicial

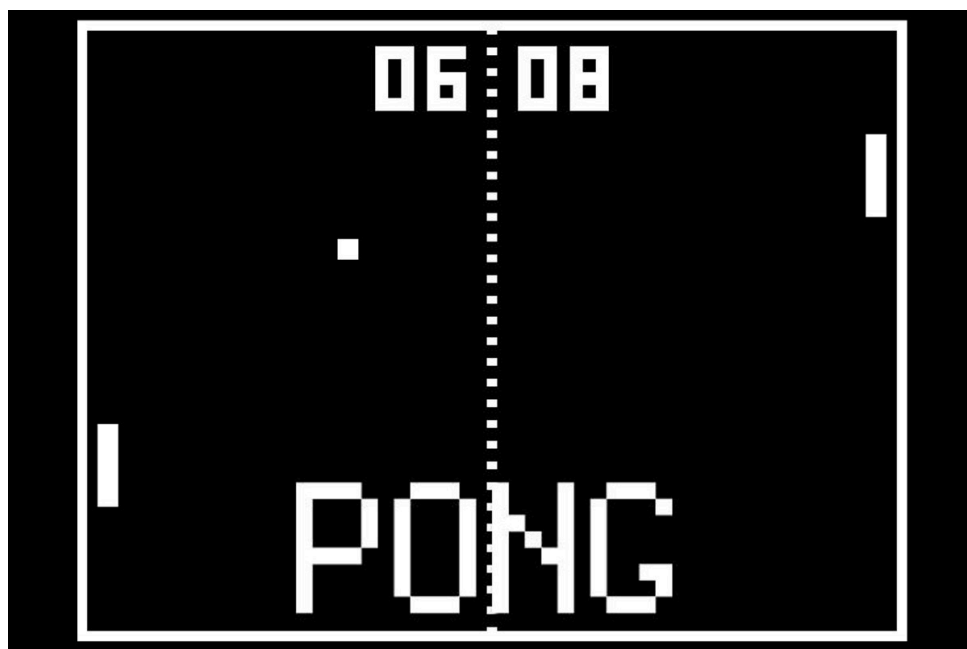
- Durante el juego

Cambios Implementados

- Sistema de Espectadores Implementación

- Desconexiones de jugadores Implementación

- Mensajes Personalizados Implementación



## Bugs

Lo único hay un problema en los espectadores y el movimiento de las palas el espectador ve el movimiento de las palas muy raro a veces no se mueven o hace flicker seguro que no estoy enviando correctamente la información al espectador

## Explicación WebSockets:

WebSockets es un protocolo de comunicación que proporciona canales de comunicación full-duplex sobre una única conexión TCP. A diferencia del modelo tradicional HTTP de solicitud-respuesta, WebSockets permite: Comunicación bidireccional: El servidor puede enviar datos al cliente sin que éste los solicite primero Conexión persistente: La conexión permanece abierta después del handshake inicial Baja latencia: Ideal para aplicaciones en tiempo real como juegos, chats, etc. El protocolo WebSocket comienza con un handshake HTTP que luego se actualiza a una conexión WebSocket. Una vez establecida, los datos pueden fluir en ambas direcciones como "frames" con muy poca sobrecarga.

## Esquema conexiones Proyecto

### Conexión inicial

Cliente se conecta al servidor WebSocket Servidor asigna rol (Jugador 1, Jugador 2 o Espectador) Servidor envía información del rol al cliente Una vez jugador 1 y jugador 2 están dentro empieza la partida

### Durante el juego

Jugadores envían su posición (coordenada Y) al servidor Servidor reenvía estas posiciones al oponente y a los espectadores Jugador 1 ("host") calcula la física de la pelota y envía su posición Servidor reenvía la posición de la pelota a todos los clientes Cuando hay un punto, Jugador 1 envía el marcador actualizado Cuando hay 3 puntos del jugador 1 o 2 se envía el finalizado de la partida Desconexión de jugadores cuando se desconectan se envían mensajes tanto a espectadores como al player 1 o 2

## Cambios Implementados

### Sistema de Espectadores Implementación

Array spectators en el servidor para gestionar conexiones Lógica para enviar datos del juego a los espectadores Mensajes especiales para espectadores para detectar si eres un espectador

### Desconexiones de jugadores Implementación

Eventos on('close') para cada tipo de conexión Notificación a los otros jugadores cuando un oponente se desconecta Establecer player desconectado a null para limpiar la variable

### Mensajes Personalizados Implementación

Para los player 1 y player 2 sale HAS GANADO / HAS PERDIDO si eres espectador le envia el numero de jugador que gano o perdio lo mismo con quien se ha desconectado