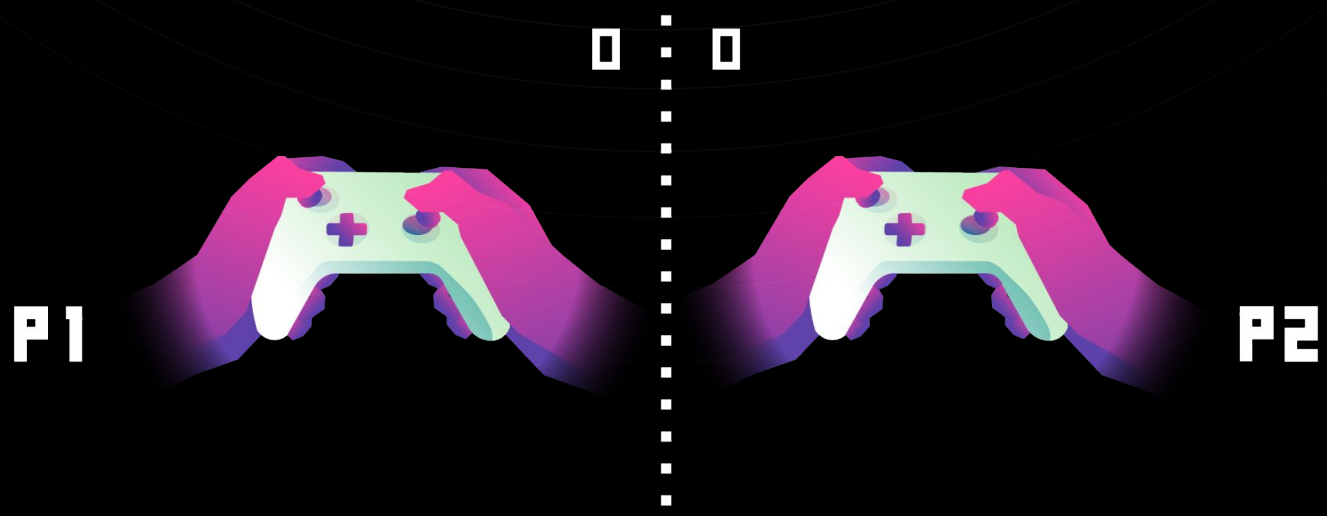
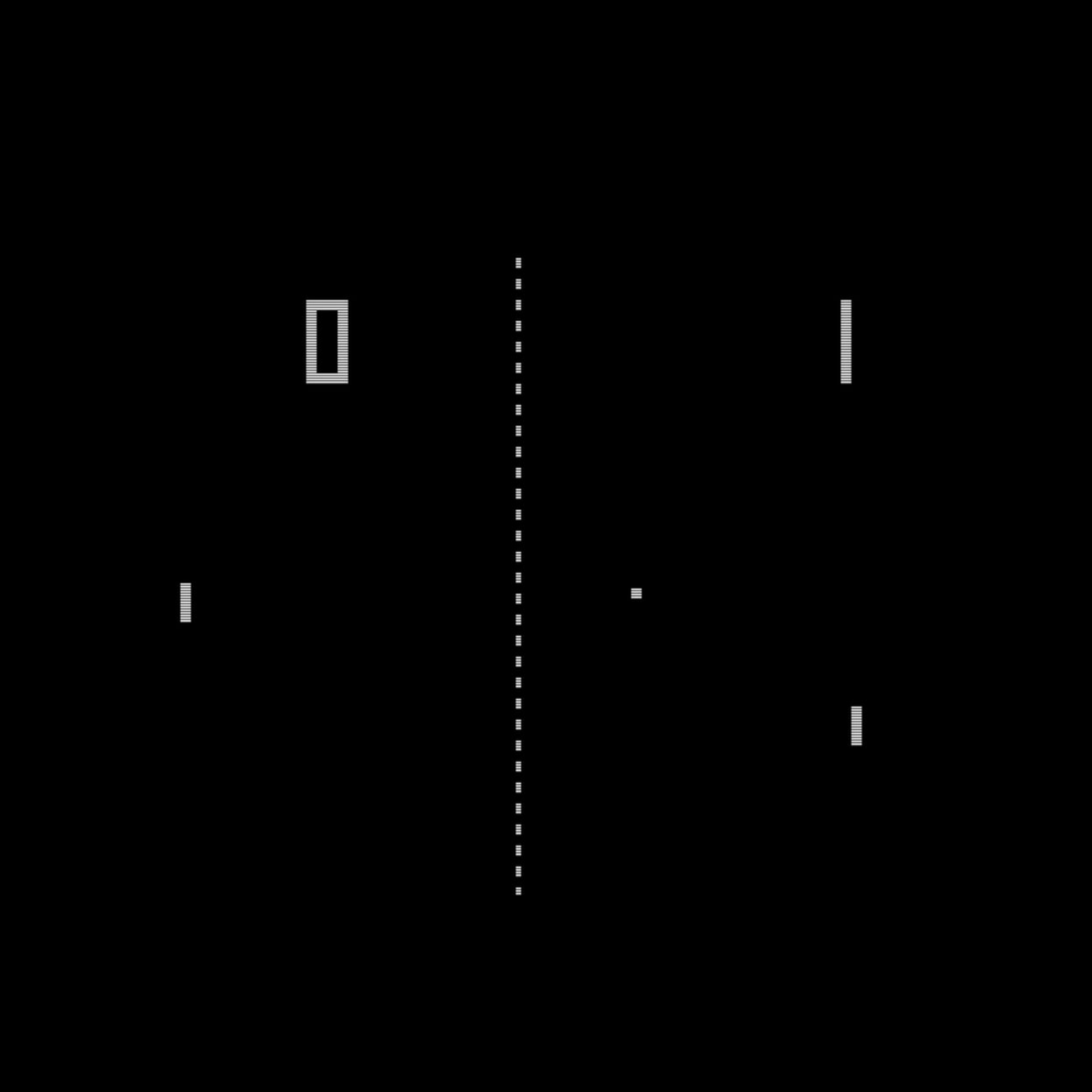
Fundamentos para

Desarrollar Videojuegos Multijugador Online

Hector Pulido
@hector_pulido_





¿Tu juego realmente necesita multijugador?

El modo multijugador NO es secundario

¿Cuándo NO necesitas un modo multijugador?

No necesitas un modo multijugador si:

- No puedes garantizar que habrá personas conectadas siempre.
- No puedes permitirte el costo extra de tener servidores.

No necesitas un modo multijugador si:

- Tu juego es frustrante cuando hay lag o latencia (AKA necesita reflejos).
- Probablemente debas hacer tu juego dos veces.

¿Cuándo SÍ es buena idea un modo multijugador?

Puede ser buena idea un modo multijugador si:

- Tu juego es inherentemente multijugador.
- Desde el principio desarrollas el modo multijugador.

Puede ser buena idea un modo multijugador si:

- Puedes doblar o triplicar tu tiempo de desarrollo.
- Tienes conocimientos de redes además de la programación de videojuegos (o terminas estos cursos :D).

Entonces… ¿Tu **juego realmente necesita multijugador**?

Consideraciones ANTES de **crear** un **juego multijugador**

1. ¿**Qué herramientas** vas a **utilizar**?

2. ¿P2P o servidores dedicados?

3. Progresión y mecánicas multijugador

4. Plataformas específicas

5. Manejo de servidor + cliente

Herramientas para **crear** un **juego multijugador**

Servidores


aws

Motor de videojuegos

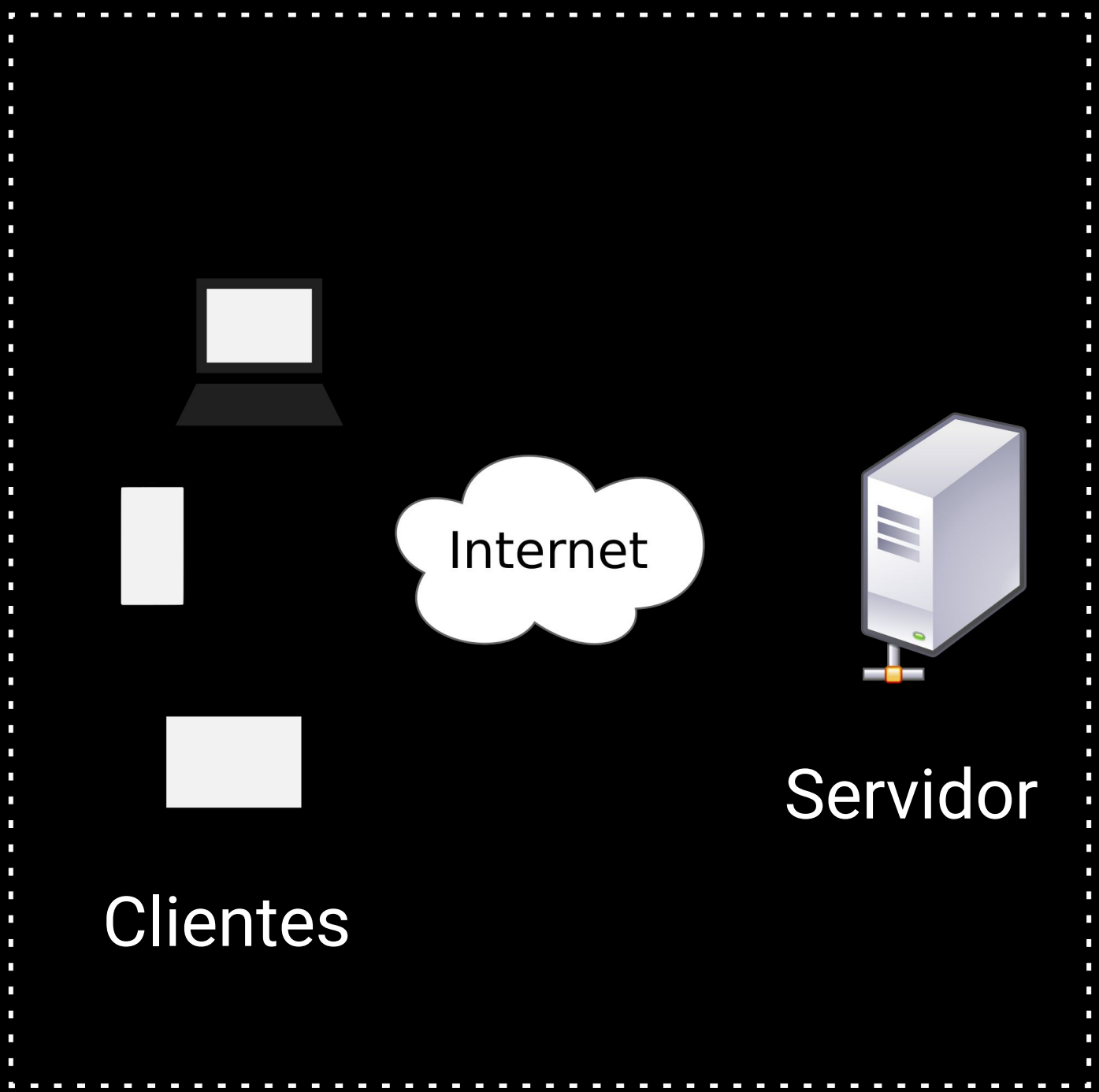
Frameworks


MIRROR


WARP


photon
CLOUD

Servidor y cliente


Internet
Servidor
Clientes

¿Qué hace el servidor?

¿Qué hace el servidor?

- Maneja datos sensibles.
- Coordina a los clientes.
- Corre una versión mínima del juego.
- Tiene la autoridad.

¿Qué hace el cliente?

¿Qué hace el cliente?

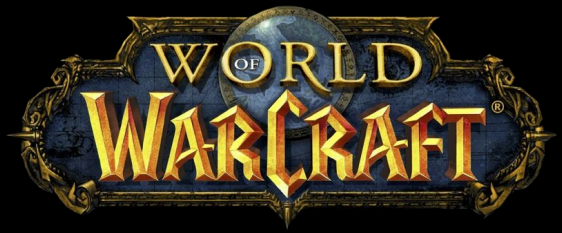
- Obtiene datos del servidor.
- Renderiza el juego para el jugador.
- Envía los inputs del usuario.

Autoridad

¿Quién tiene el control en tu juego?

Mínima autoridad
(juego local)

BATTLEFIELD™



Máxima autoridad

Problemas de pobre autoridad

Problemas de pobre autoridad

- Vulnerabilidad a hacks.
- Alta entropía en la sincronización.
- Mayor cantidad de situaciones frustrantes.

Demasiada autoridad tampoco es buena

Problemas de demasiada autoridad

- Lag.
- Mayor ancho de banda.
- Muchos más posibles puntos de fallo en el código.

P2P vs. servidores dedicados

Ventajas de P2P

Ventajas de P2P

- No se necesita un servidor central.
- Alta escalabilidad.
- Sistema CASI gratuito.

Desventajas de P2P

Desventajas de P2P

- Estabilidad nula.
- Es muy difícil evitar trampas.
- Puede requerir *port forwarding*.

Ventajas de servidores privados

Ventajas de servidores privados

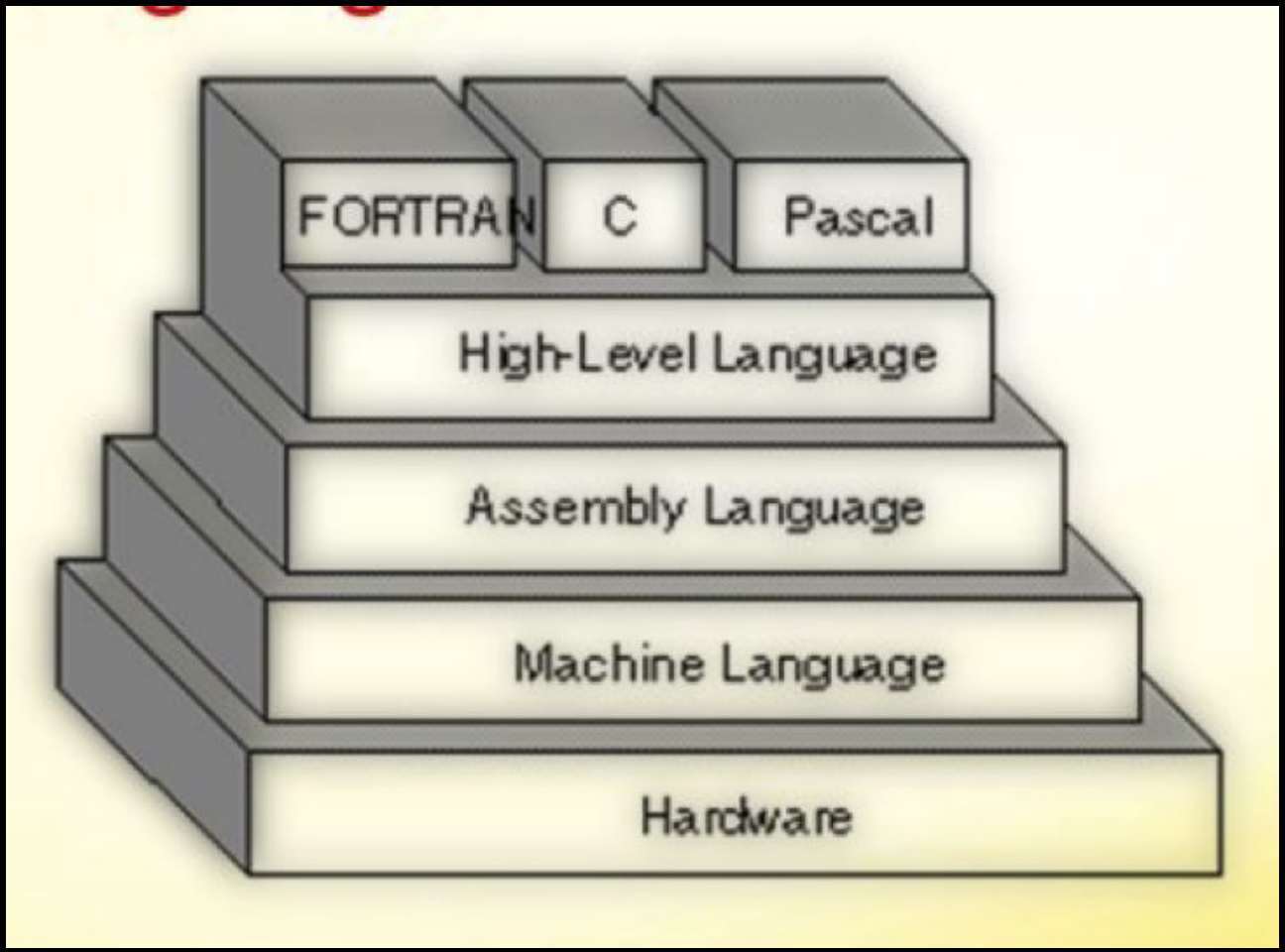
- Fácil de implementar.
- Mejor estabilidad y control.
- Máxima autoridad.

Desventajas de servidores privados

Desventajas de servidores privados

- Dinero.
- Si el servidor tiene problemas...

High level o low level


FORTRAN
C
Pascal
High-Level Language
Assembly Language
Machine Language
Hardware

¿Por qué elegir sockets?

Ventajas de sockets

- Control de cada aspecto.
- Eficiencia de red y recursos.
- Sin costos ocultos.

¿Por qué elegir frameworks?

Ventajas de frameworks

- Simplicidad al máximo.
- Documentación y soporte.
- Abstracciones de alto nivel.
- Nubes plug & play.

Frameworks populares


GAMESPARKS


photon
CLOUD


MIRROR

Muchos más...

¿Qué es Mirror? ¿Por qué vale la pena?

Ventajas de Mirror

- Compatibilidad con UNET.
- Código abierto.
- Cero costos ocultos.

Ventajas de Mirror

- Modularidad.
- Server y cliente en el mismo lugar.

Próximos pasos