

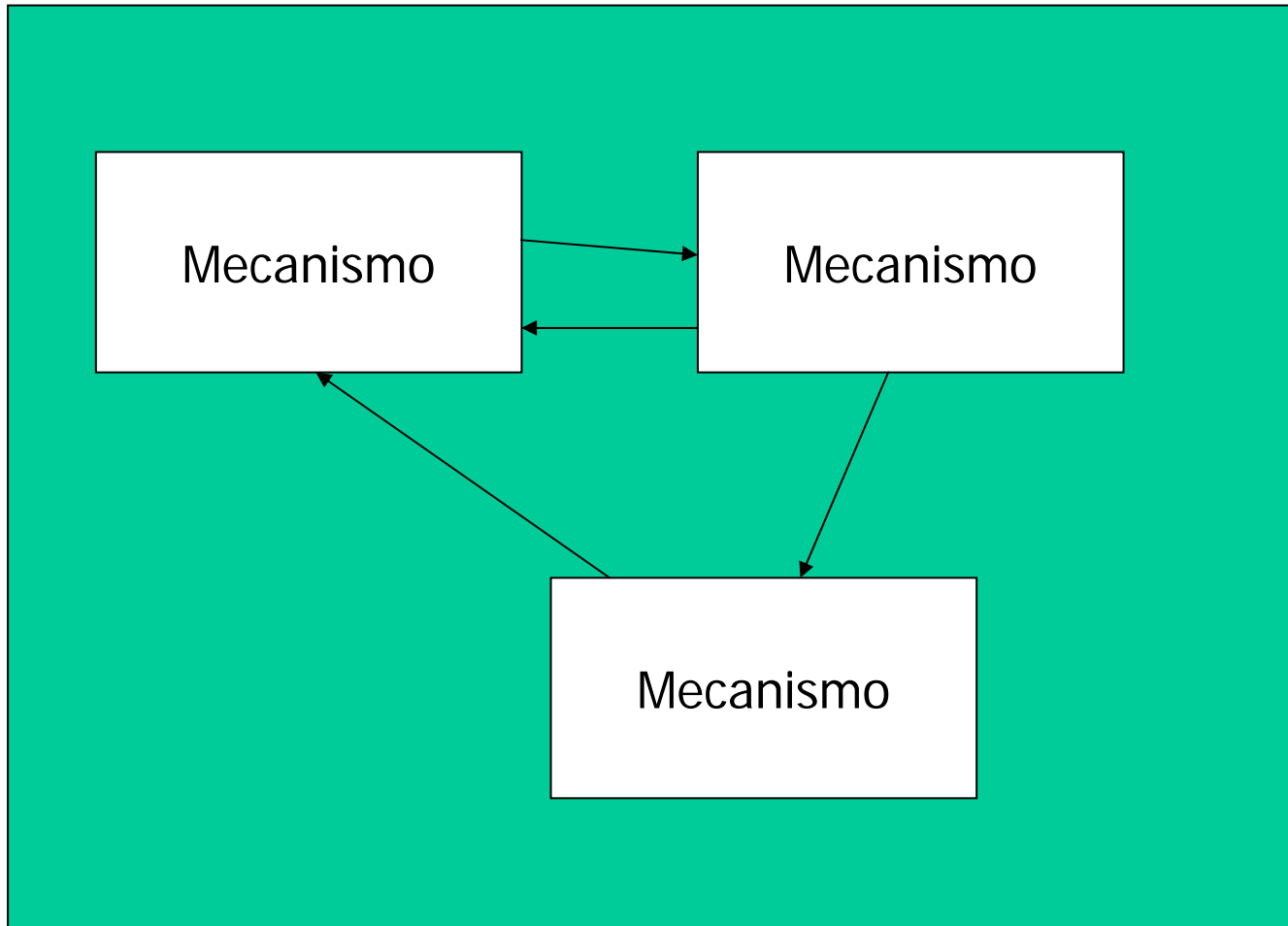
Diseño - Mecanismos



Departamento de Ingeniería del Software e Inteligencia Artificial
Universidad Complutense de Madrid

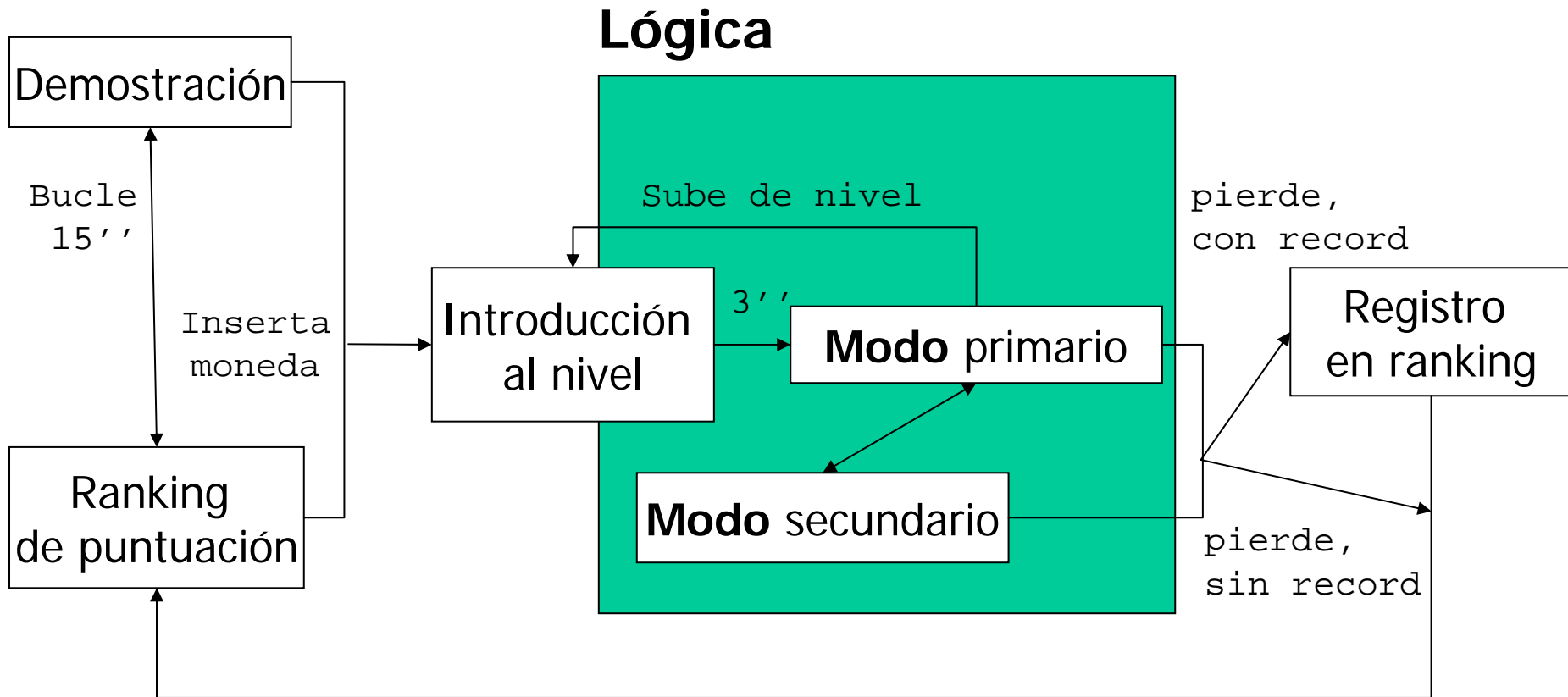


Sistema



- ❑ *Mecánica*: Aparato o resorte interior que da movimiento a un ingenio o artefacto (El término más usado en la industria)
- ❑ *Mecanismo*: Estructura de un cuerpo natural o artificial, y combinación de sus partes constitutivas
- ❑ Nuestra definición: **modelo** de una **oportunidad de interacción** del jugador con el juego
 - ❑ **Modelo**: especificación formal de un subsistema del juego lo bastante completa como para comenzar a proceder a su implementación
 - ❑ **Oportunidad de interacción**: contexto + acciones posibles asociadas a una tarea concreta

- ❑ Básicamente consiste en definir ¿a qué juega el jugador?
- ❑ Distinción entre *in-game* y *out-game*
 - ❑ **In-Game**: desarrollo de la *sesión de juego* propiamente dicha
 - ❑ *Modos de juego* primarios y secundarios
 - ❑ **Out-Game**: modos de ejecución que no forman parte de dicha sesión
 - ❑ Si no se usan con precaución pueden arruinar la inmersión
 - ❑ Depende de lo que el diseñador entienda que forma parte del juego
 - ❑ ¿Creación de un personaje, configuración de un ejército...?
- ❑ Interacción implica **interfaz**



Un **modo de juego** tiene estos componentes:

- Un conjunto de modos de percepción
- Un conjunto de mecanismos
- Retos y acciones característicos de su *jugabilidad*
- Hay modos de juego primarios y secundarios
- En un videojuego hay muchos modos de ejecución, pero no todos son modos de juego



- ❑ Proponemos una *secuencia ideal*, que luego matizaremos
- 1. Definición de objetivos
 - ❑ No basta con saber que “será divertido”, expresa el objetivo en términos de lo que sentirá el jugador
 - ❑ Es un proceso subjetivo, la forma de ver el mundo de los artistas
- 2. Definición de temas
 - ❑ El entorno concreto donde se va a perseguir el objetivo
 - ❑ El tema es el que va **subordinado al objetivo**, y no al revés
- 3. Investigación y preparación
 - ❑ Ajustar los objetivos
 - ❑ Trabajar “en papel”, sin implementar nada

4. Fase de diseño integral

1. Interfaces entrada/salida

- Es la más importante: define lo que se puede y lo que no se puede hacer

2. Estructura del juego

- Da forma a los ideales del juego
- Definir el modelo del mundo, su nivel de manipulación
- Definir el elemento clave (relacionado con el género), puede haber combinaciones, pero no deben *ensuciar* el juego

3. Arquitectura general del programa

- Arquitectura software, diseño de los motores, etc.

5. Evaluación del diseño

- Primeras impresiones: ¿es lo que yo quería para mis jugadores?
- Probar la estabilidad / robustez del juego
- Probar la dificultad del juego (que no existan *atajos* imprevistos)
- Luz verde** o **luz roja**

6. Documentación del juego

7. *Implementación del juego*

8. Pruebas (*alpha* + *beta*)

- Redefine el diseño si fuera necesario
- Aprovecha para configurar los parámetros que has dejado libres
- Si hallas un error grave, repite el diseño; si un error fatal, **luz roja**
- No muestres el juego a los *betatesters* hasta que no esté preparado

9. Publicación

- ❑ La documentación final es tarea de todos: artistas, diseñadores y programadores

10. Mantenimiento

- ❑ Especialmente MMOG o juegos muy configurables

11. Post-Mortem

- ❑ Lloverán las críticas, los jugadores harán lo impensable con tu juego, no se leerá nadie las reglas, etc.
- ❑ Es importante que llegue al público: es una obra de arte
- ❑ **Es más importante que te guste a ti:** es *tu* obra de arte

❑ Dosis de realismo

- ❑ No es una secuencia, suele haber iteraciones
- ❑ Diseño e implementación se solapan
 - ❑ Se implementan prototipos desde el principio
- ❑ Diseño y tecnología están acoplados
 - ❑ A menudo en las primeras fases de diseño se trabaja sobre una tecnología inacabada
- ❑ Los ajustes se hacen especialmente a última hora
- ❑ La luz verde y la luz roja no son cosa tuya
- ❑ ¡En la práctica lo importante es *que venda!*



- ❑ Especificar rigurosamente *¿a qué quiero que juegue el jugador?*
- ❑ Mejor pocas reglas y buenas, que muchas y confusas
 - ❑ **Un diseño es perfecto no cuando no hay nada que añadir, sino cuando no hay nada que quitar**

The figure of merit for a good game is Minimum Rules, Maximum Richness. Nolan Bushnell (fundador de Atari)

- ❑ Ser consciente del coste y la viabilidad de cada mecanismo
- ❑ Buscar la **ortogonalidad** (independencia entre características)
- ❑ Evitar las excepciones
- ❑ Sistemas autónomos complejos: ¡no puedes predecir al jugador, ni lo que va a *emerger* de tus mecanismos!
 - ❑ No existe verificación formal, hazlo **robusto** y **pruébalo mucho**

- Equilibrio del sistema
- Relaciones entre oponentes
- Aprendizaje
- Ilusionismo



- ❑ El jugador y la máquina son criaturas diferentes
 - ❑ Tratar de hacer *más humano* al ordenador para que resulte un oponente adecuado
 - ❑ Lo que puede significar mejorar o “empeorar” su rendimiento
- ❑ Para crear oposición en juegos monojugador pueden utilizarse estas técnicas:
 - ❑ Superación en recursos
 - ❑ Inteligencia artificial
 - ❑ Control de la información
 - ❑ Cambios de ritmo

- ❑ El número de oponentes es mayor o sus características muy superiores a las de los jugadores
- ❑ Esto permite superar al jugador sin necesidad de reproducir inteligencia
- ❑ Ventajas
 - ❑ Produce enfrentamientos del tipo *David contra Goliat*
 - ❑ Fácil de implementar
- ❑ Inconvenientes
 - ❑ Se usa en demasiados juegos, ¡incluso actuales!
 - ❑ No da sensación de competición realista, justa ni equilibrada

- ❑ Se estudian autómatas, agentes autónomos o sistemas expertos de gestión
- ❑ Ventajas
 - ❑ Evita comportamientos estúpidos o ilógicos, que son frecuentes
 - ❑ Puede ser más difícil de predecir
 - ❑ Se puede parametrizar / configurar con *scripts*
- ❑ Inconvenientes
 - ❑ Consumo de recursos y programación más compleja
 - ❑ Puede que la IA resulte *demasiado perfecta* y haya que trugarla (simular errores e imprecisiones naturales es difícil)

- ❑ Ocultar información al jugador o dosificar su entrega
 - ❑ Análogo a la *dosificación del argumento* en una historia
- ❑ Proporcionar información extra a los oponentes
- ❑ Ventajas
 - ❑ Fácil de implementar
 - ❑ En muchos juegos esto tiene sentido, ambienta y estimula al jugador
 - ❑ Ejemplo clásico: la niebla de batalla
- ❑ Inconvenientes
 - ❑ Falta de realismo, sensación de falsedad

- ❑ Aprovechar la velocidad de cálculo del ordenador y acelerar el juego, forzando la acción cada cierto intervalo de tiempo
 - ❑ Cronómetro explícito / implícito
- ❑ También se puede hacer más compleja la misión del jugador, ej: asignándole demasiadas tareas simultaneas
- ❑ Ventajas
 - ❑ Fácil de implementar
 - ❑ Fácil de configurar y ajustar el nivel de dificultad
- ❑ Inconvenientes
 - ❑ Se lleva usando desde los primeros juegos
 - ❑ No da sensación de ser un reto justo
 - ❑ Sólo tiene sentido en los juegos de acción



- ❑ El enfrentamiento o conflicto es la clave del juego, define el tono del juego y buena parte de su mecánica
- ❑ Distinguimos estos tipos de relaciones:
 - ❑ Relaciones simétricas
 - ❑ Relaciones asimétricas
 - ❑ Relaciones de triangularidad
 - ❑ Relaciones indirectas y a través de actores

- ❑ Ambos oponentes están en igualdad de condiciones
- ❑ Ventajas
 - ❑ Equilibrio garantizado para juegos deportivos o multijugador
 - ❑ Fáciles de programar (puedes reusar código del oponente)
 - ❑ Fáciles de aprender y utilizar
- ❑ Inconvenientes
 - ❑ Resultan simples, poco útiles para juegos realistas
 - ❑ Puede que la estrategia pierda valor, si puede realizarse por ambos oponentes al mismo tiempo
 - ❑ Gana valor la táctica, la forma de llevar a cabo la estrategia
- ❑ Las *relaciones simétricas funcionales* son también simétricas aunque dan la impresión estética de no serlo

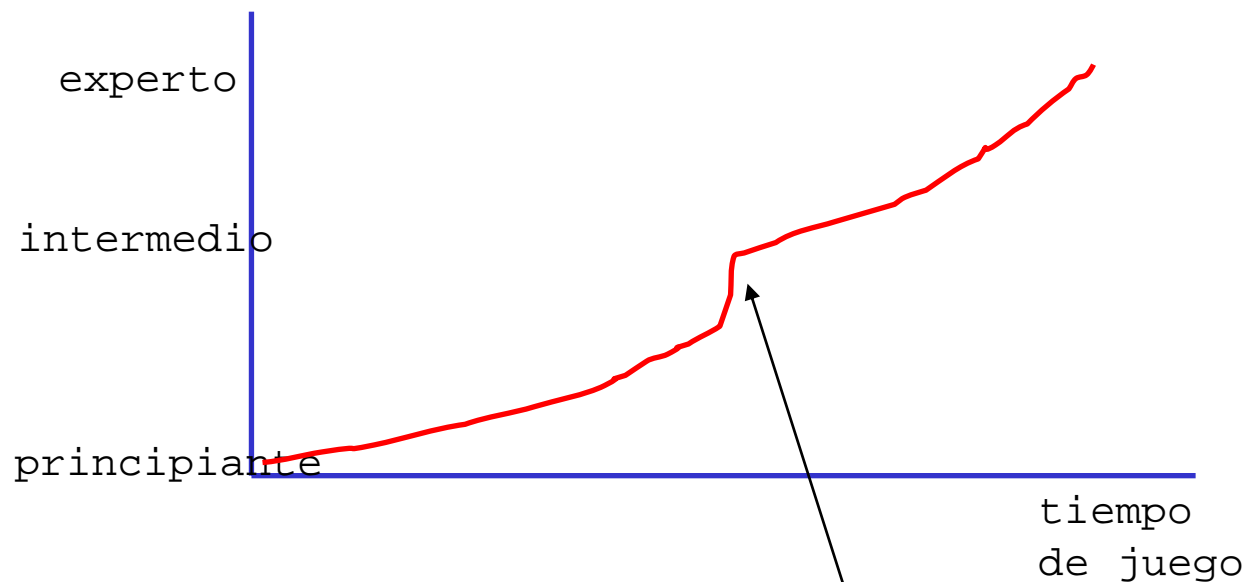
- ❑ Cada oponente tiene unas características distintas, pero mantienen las mismas posibilidades de victoria (equilibrio)
- ❑ *Asimetría plástica*: parte de una simetría formal que se reparte proporcionalmente en función de las elecciones de los jugadores
- ❑ *Transitividad*: permite definir estrategias cada vez mejores pero más costosas, que vencen a todas las anteriores
- ❑ Ventajas
 - ❑ Aportan riqueza y variedad al juego
 - ❑ En algunos juegos la asimetría pura y dura es deseable
- ❑ Inconvenientes
 - ❑ La transitividad resulta predecible y sólo sirve en juegos que progresan hacia un final definitivo
 - ❑ Son algo más difíciles de equilibrar

- ❑ La triangularidad permite incluir relaciones *no transitivas* entre los elementos del juego
 - ❑ Ej: *Piedra, papel o tijeras*
- ❑ En la práctica en vez de tres elementos, se usan mezcla de acciones *ofensivas-defensivas*
- ❑ Ventajas
 - ❑ Aportan riqueza y variedad a la interacción del juego
- ❑ Inconvenientes
 - ❑ Interacción más compleja y difícil de aprender
 - ❑ Se restringen algunas acciones de los jugadores

- ❑ La indirección es la esencia de la triangularidad
 - ❑ No es más que generalizar la triangularidad a N dimensiones
- ❑ Incluir *actores* (pasivos o activos) y permitir que la interacción con ellos sea a veces competitiva y a veces cooperativa
 - ❑ Todo se vuelve más interesante si los costes ocultos de las estrategias no son estáticos sino que varían en el tiempo
- ❑ Ventajas
 - ❑ Estos conflictos pueden modelar conflictos reales, más complejos y menos violentos que el *uno-contra-uno*
- ❑ Inconvenientes
 - ❑ Interacción más compleja y difícil de equilibrar
 - ❑ Difícil de aprender para el jugador



- ❑ No se crea un solo juego, sino una familia de juegos relacionados de dificultad creciente
 - ❑ Principiante – Intermedio – Experto
- ❑ La curva debe ser *suave*



truco

Hay que tratar de evitarlo



- ❑ El juego debe mantener la *motivación* en todo momento
- ❑ Ilusión de “ganabilidad”: El éxito debe parecer alcanzable para el principiante, para el jugador medio y para el experto
 - ❑ **Más allá**: Ilusión de “ganador”: el éxito debe SER alcanzable fácilmente al tiempo que parece difícil
- ❑ De nuevo los juegos limpios ayudan a crear esta ilusión en el jugador
- ❑ **Pruebas del ilusionista**
 - ❑ El jugador debe echarse las culpas a si mismo por sus fracasos, y no al juego o al diseñador
 - ❑ El jugador debe sentirse más amenazado de lo que en realidad está
 - ❑ ¡Los testers no deben ser conscientes de la ilusión!

- ❑ Adams, E.: Designing With Gameplay Modes and Flowboards. Gamasutra, Designer's Notebook (2004)
- ❑ Crawford, C.: Chris Crawford on Game Design. New Riders (2003)
- ❑ Meigs, T.: Ultimate Game Design: Building Game Worlds (2003)
- ❑ Mulligan, J; Patrovsky, B.: Developing Online Games: An Insider's Guide (2003)
- ❑ Rollings, A., Adams, E.: On Computer Game Design. New Riders 1st edition (2003)
- ❑ Salen, K.: Rules of Play : Game Design Fundamentals (2003)

Federico Peinado
<http://federicopeinado.com>