Diseño – Equilibrado (Preparación)

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

- **Equilibrar** = hacer que el factor determinante del éxito sea *principalmente* la habilidad del jugador
- Cada género tiene un equilibrado distinto, con aspectos comunes
  - RTS no es el más complicado porque se puede usar teoría militar
- El equilibro parece ser un concepto *relativo al jugador*
  - Suponer un *jugador medio* como público objetivo
  - ¡No elegir como jugador medio a uno mismo!
- Los fracasos por desequilibrio no son fáciles de detectar
- Maestros del equilibrado de un videojuego
  - Shigeru Miyamoto (Donkey Kong, Mario, Zelda)
  - Brian Reynolds (Civilization II, Alpha Centauri, Rise of Nations)
  - Sid Meier (Railroad Tycoon, Civilization III, Pirates!)

- Fácil
  - Podrían existir niveles “muy fáciles”, incluso niveles absurdos donde los enemigos huyan de miedo… (depende de nuestra filosofía)
  - Los *trucos* son otra forma de bajar el nivel, diséñalos también
    - Si el diseñador no lo contempló es *trampa*, lo cual es sinónimo de *fallo en el diseño* ya que no tendrían que permitirse trampas
- Normal (Fácil + 15%)
  - El que proporciona la experiencia deseada al jugador medio
  - Idealmente el jugador medio no sufriría fracasos estériles
- Difícil (Fácil + 30%)
  - Podrían existir niveles “muy difíciles” (Fácil + 50%) aunque nunca imposibles, pensados para *hardcores* o fanáticos del juego
  - No es justo que se guarden sorpresas exclusivas sólo para este nivel

- Diseñad también trucos “tramposos” de cara a poder analizar/depurar el juego “en marcha”
  - La mayor parte de estos accesos formarán parte de la versión *degub* pero NO de la versión definitiva
  - TAMBIÉN deben ser bien diseñados... ahorrar un sólo segundo por operación de depuración afectará significativamente al proyecto
- En la versión definitiva sí podéis añadir sorpresas y curiosidades orientadas al jugador, con contenido extra del juego, atajos, vidas infinitas y cosas así
  - ¡No perdáis mucho tiempo con esto!
  - Ej: Llamar al personaje ZELDA en el Zelda de NES

- El sistema que equilibra el juego puede o no ser visible para el jugador (ej. Nevewinter Nights)
- Intratable formalmente
  - Problema de optimización n-dimensional con valores continuos
- **Prueba y error**, con diferencia la única técnica más utilizada
  - Incluso llegando a añadir parches para corregir problemas de equilibrio
  - Método costoso en tiempo y recursos, muy propenso a errores
- En el futuro tal vez se puede automatizar parte del proceso
  - Herramientas de prototipado rápido (como Blitz3D)
    - Distinguir entre el código final y el código del prototipo
  - ¿Ajuste estadístico automático aprovechando Internet?

- Tradicionalmente se hablaba de equilibrio en las aperturas, el medio juego y los finales
- **Equilibrio estático**
- **Equilibrio dinámico**

- Relación (invariante en el tiempo) de las reglas del juego entre sí
- Establecer valores de referencia y diseñar todo en relación a estos
  - Unidad de longitud (longitud de plataformas, potencia de salto…)
  - Unidad de fuerza (fuerza de cada tropa, resistencia de los edificios…)
- Ejecutar automáticamente baterías de partidas y calcular promedios
- Usar matrices de resultados para comparar estrategias
  - Si hay elementos aleatorios o no modelables: dar una puntuación "aproximada" a cada partición discreta del espacio de posibilidades

Ej. Acción\Contexto	Es vuestro aniversario	No es vuestro aniversario
La regalas flores	10	20
No la regalas flores	99	50

***** Probabilidad (%) de discutir con tu mujer

- **Estrategia dominante *fuerte*** = (casi) siempre ganas
  - No debería existir en ningún juego serio de competición (Akuma se prohibe en campeonatos de *Turbo Street Fighter 2*)
  - Producir warlocks en *Warcraft*, producir tanques (tank rush) en *Red Alert,* o chutar desde la línea del área en *Euroleague*
- **Estrategia dominante *débil*** = (casi) nunca pierdes
  - En algunos casos puede permitirse

- Existen **costes ocultos** difíciles de estimar (ej: asigna un precio para las flores y haz la conversión a unidades de afecto)
- La **combinación** no debería dar problemas si las unidades atómicas son independientes y están equilibradas
- La **emergencia** debe usarse con conocimiento
- No abusar de la **aleatoriedad**
- Los **ciclos de realimentación** deben cortarse a tiempo para que no desequilibren
  - “El rico se enriquece más y el pobre se empobrece más”

- Equilibrado pasivo vs. Equilibrado activo
- En realidad el juego es un sistema (idealmente en equilibrio) que cambia con el paso del tiempo y las acciones de los jugadores
  - El objetivo es mantener una cierta coherencia y orden interno
  - Este equilibrio es lo que aporta la verdadera jugabilidad al juego
  - En cada modo de interacción hará falta un equilibrado
- Modos de interacción que afectan al equilibrio
  - El jugador debe mantener el equilibrio (ej. Tetris)
  - El jugador debe restaurar el equilibrio (resolver un puzzle)
  - El jugador debe destruir el equilibrio (acabar con todos los enemigos)

- Característica que mejora la jugabilidad facilitando el ajuste del juego, aunque a un precio
- Aproximaciones “económicas”
  - Juegos de coches, basandose en la posición que ocupas
  - No usar DDA, FIFA: cambio manual de dificultad en medio del partido
- Aproximaciones algo más trabajadas
  - Final Fantasy VIII: la vida de los enemigos se multiplica por el promedio de nivel de tus personajes / 10
  - El nivel de los bots en Unreal Tournament se ajusta según tu precisión y tus frags
  - Max Payne 2: Casi nadie se molestaba en cambiar el nivel de dificultad, y como querían lucir la inteligencia de los bots, implementaron DDA
  - En SIN Episodes aparecen enemigos con casco si disparas a la cabeza

- Proporcionar un reto constante
  - Buena curva de aprendizaje
- Proporcionar la “sensación de reto justo” al jugador
  - Ninguna decisión tomada sin la debida información debe condenar al fracaso al jugador (Monty on the Run, Deathtrap Dungeon)
  - No deben ser necesarias ni guías ni ayudas externas al juego
  - Tratar bien a los novatos (tutoriales, areas especiales...)
- Evitar el estancamiento
  - Dar realimentación positiva o negativa
- Evitar tareas triviales o innecesarias
  - Ej. Alpha Centaury permite automatizar casi toda la gestión

- Permitir variar el nivel de dificultad (si procede)
  - Adaptación automática (Max Payne) o mediante preguntas contextuales (Leisure Suit Larry: Magna Cum Laude, Pirates!)
  - John Laird demostró que para hacer más inteligentes los bots del Quake la mejor estrategia es disminuir su tiempo de reacción
- Renunciar al realismo si el equilibrio del juego lo requiere
- Diseña pensando en modificar después
  - Reglas nucleares en código
  - Parámetros variables en datos
- Diseña también las pruebas y los experimentos
  - Parámetro a parámetro, de forma metódica
  - Variando significativamente el valor, buscando por divide y vencerás en el rango de valores

- Daniel Sánchez Crespo
- GUEIM
  www.gueim.org
- Juan deMiguel Contreras

Federico Peinado
http://federicopeinado.com