



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Producción de un videojuego

Ramón Mollá

Dpto. Sistemas Informáticos y Computación

UPV

Objetivos de aprendizaje

Conocer el ciclo de desarrollo de un videojuego

Presentar las diferentes fases del ciclo de vida haciendo especial hincapié en la investigación y preparación de los tópicos y los objetivos del juego

Conocer algunas implicaciones y aspectos básicos del diseño del videojuego

Conocer la tecnología Push

Índice

Introducción

Fases del ciclo de vida

Investigación y preparación

Tópicos y Objetivos

Diseño

Evaluación del diseño

Pre y programación

Testeo

Pulido

Manual

Push

Post-mortem

Introducción

La especificación de un juego es variopinta

Depende de presupuesto, complejidad, tipo,...

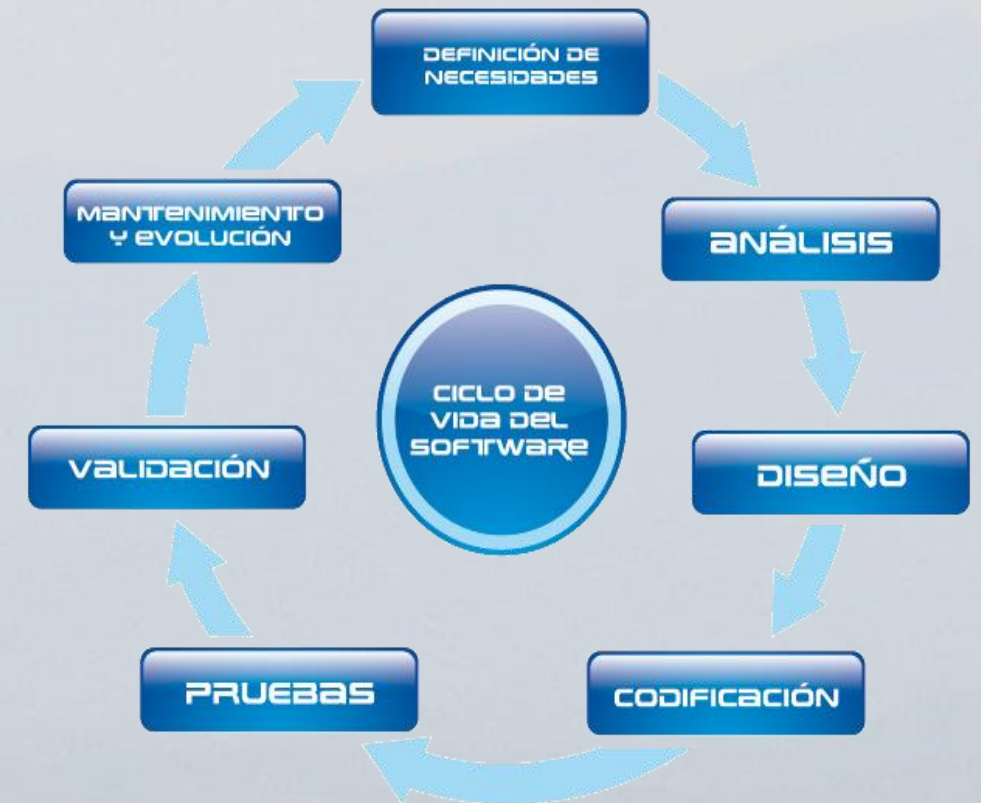
Proceso complejo difícilmente normalizable

Normas flexibles o guías generales

No encorsetarse rígidamente

Mezcla de arte y tecnología. Balance difícil

Sigue el ciclo de vida de una aplicación informática



Fases del ciclo de vida

Investigación y preparación

Diseño

Evaluación del diseño

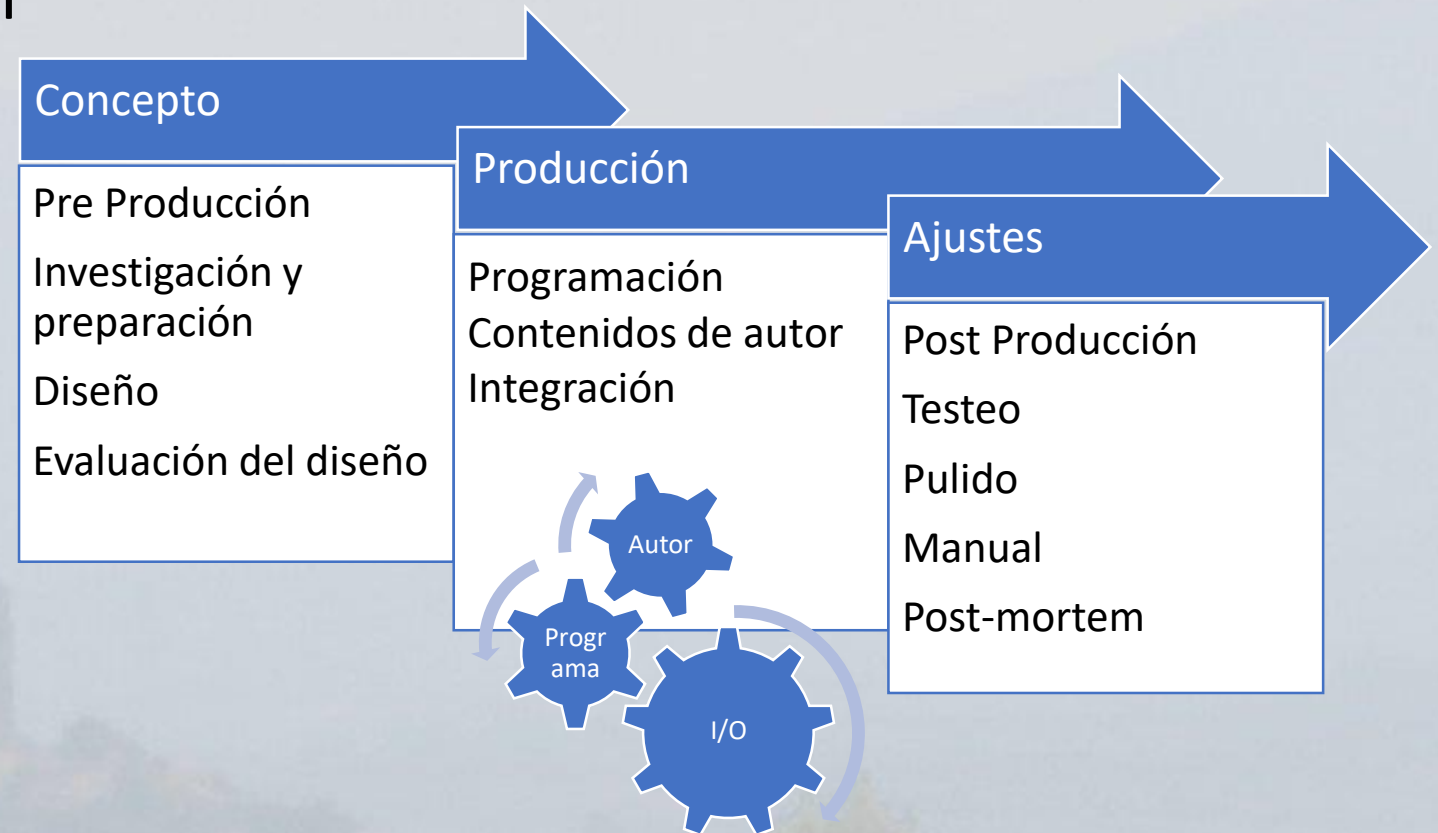
Pre y programación

Testeo

Pulido

Manual

Post-mortem



Fases del ciclo de vida

Investigación y preparación

Diseño

Evaluación del diseño

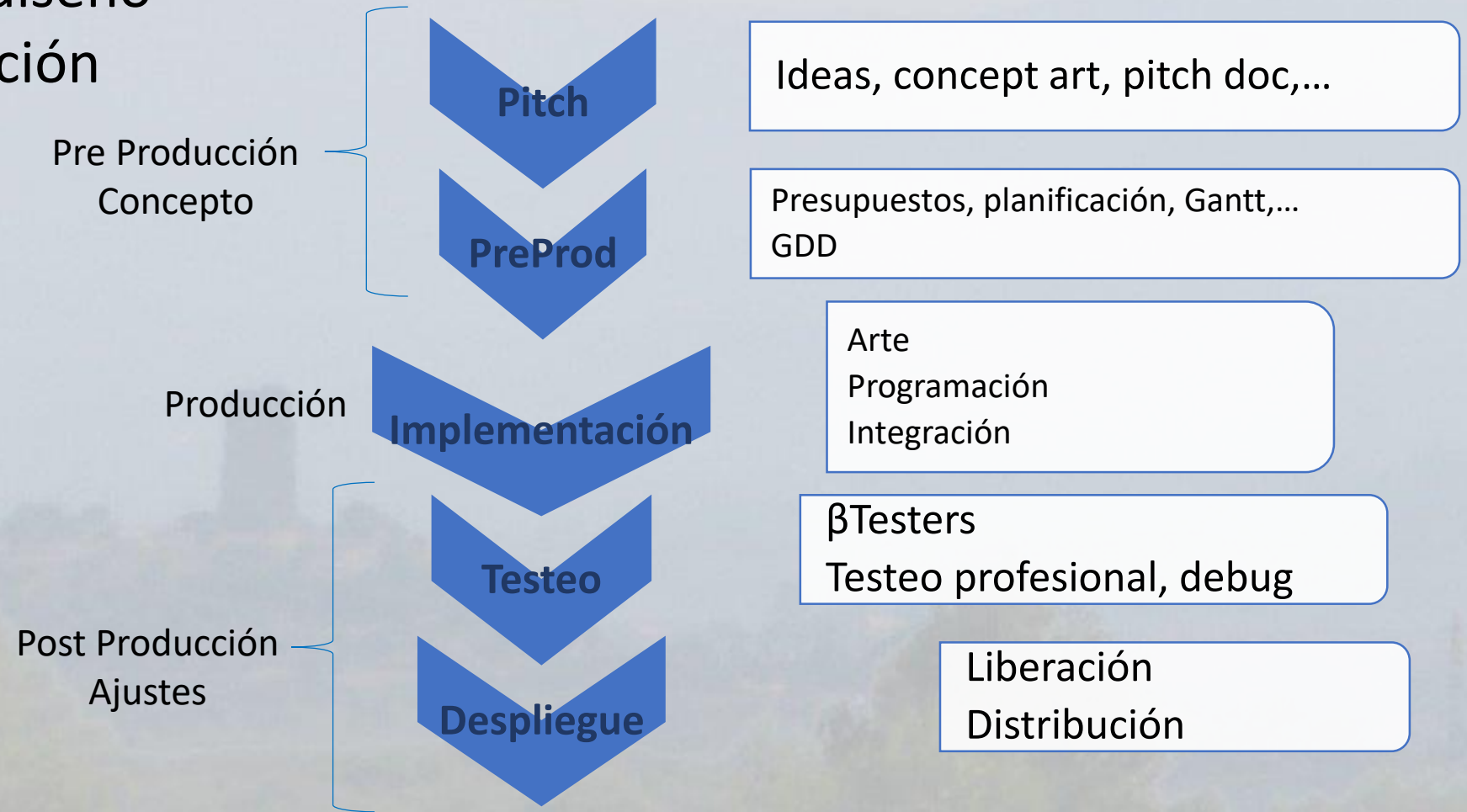
Pre y programación

Testeo

Pulido

Manual

Post-mortem



Investigación y preparación (I)

Documentación acerca de

- Objetivos. ¿Qué habilidad se pretende desarrollar en el jugador?
- Tópico. Fantasía que desarrolla los objetivos
- Referentes culturales, aportaciones previas, creencias populares,...

Descubrimiento de nuevo material puede

- Inutilizar tópico
- Cambiar objetivos. Si esto ocurre... CUIDADO. Marcador que indica que objetivos no están claramente definidos



Investigación y preparación (II)

Durante esta fase, todavía *no se crea ni una línea de código*

Puede durar meses

Uso de tormentas de ideas, lista de especificaciones, estudios de mercado,...

Visualizar todo integrado, plantear situaciones, escenarios de juego (no fondos de pantalla)

No impaciencia

Precipitación en esta fase puede generar errores en fases posteriores



Investigación y preparación (III)

Selección de objetivos (I)

Deben ser

- Claros y concisos

- Excitantes y/o divertidos

Debe definirse

- Fantasía que desarrolla el juego (metáfora)

- Tipo de emoción que desarrolla el jugador

- Aspecto educacional. Qué aprenderá el jugador



Ejemplo

Fantasía: Epic, un mundo secreto

<https://www.youtube.com/watch?v=liBYgAXGbaE>

¿Emoción desarrollada? ¿Aspecto educacional?



Investigación y preparación (IV)

Selección de objetivos (II)

Es muy subjetiva

Refleja la ética, estética y cosmovisión del diseñador

Coherentes entre lo que el público desea y lo que el diseñador cree, le interesa o le apasiona

Si interés, creencias, experiencias o pasión del diseñador no coincide con los del juego → juego mediocre y poco creíble

Deben entroncar con experiencias previas de jugador, sus expectativas y pasiones

Situaciones en que esto no es posible: FPS, espacial, fantástico,...



Investigación y preparación (II)

Selección de tópicos (III)

Tópico

Forma de expresar un objetivo

Entorno o marco de desarrollo del juego

Ej: combate espacial, pasadizos lúgubres, plataformas móviles, guerrilla en la jungla, conducción de coches, cocina hotel,...

Puede que inspiración inicial del juego parta de un tópico

Nivel de **tópico** siempre por debajo de **objetivos** a cubrir

Tópico siempre al servicio del objetivo

El tópico es lo accidental

No mueve los objetivos



Investigación y preparación (II)

Selección de tópicos (IV)

Tópico

Varios tópicos candidatos para el mismo objetivo

Seleccionar el que más se adecue al objetivo

Tarea ardua

Carga emocional del tópico puede interferir con el objetivo

Ejemplo

Objetivo: desarrollar los reflejos

Tema o tópico: carreras de cuadrigas, lanzadores de pelotas de tenis que haya que devolver, recoger productos de una cadena de producción,...

Problema: Carreras son muy lentas para satisfacer objetivos

Sugerencia: cambiar de objetivos o cambiar el tipo de carreras a carreras de coches, aviones o pod racers (star wars)



Investigación y preparación (II)

Selección de tópicos (V)

Ejemplo 2

Objetivo: Aprender a escribir a máquina

¿Tema o tópico?

Flota alienígena de naves que llevan cada una letra y hay que “matarla” pulsando la letra correspondiente sin mirar el teclado. Rol: Artillero de un cañón láser



Investigación y preparación (II)

Selección de tópicos (VI)

Ejemplo 2

Objetivo: Aprender a escribir a máquina

¿Tema o tópico?

Coche de carreras que avanza más rápido cuanto más rápidamente y sin errores escribes un texto que aparece en pantalla. Rol: Piloto de carreras



Investigación y preparación (II)

Selección de tópicos (VII)

Ejemplo 2

Objetivo: Aprender a escribir a máquina

¿Tema o tópico?

Cinta transportadora que lleva platos con letras. Distribuirlos según se van pulsando en el teclado. Rol: Camarero



Investigación y preparación (III)

En la fase de **Preparación**, se determinan

Las herramientas y decisiones necesarias para configurar el entorno de desarrollo

La especificación del sistema objetivo y el mercado objetivo. Los hitos en preparación consisten en:

Selección de equipos. Generar una matriz de todos los roles requeridos durante cada fase del desarrollo del juego

Especificaciones del sistema objetivo. Determinar los dispositivos que utilizarán los usuarios finales

Entorno de desarrollo: configuración de las herramientas y tecnologías que usará el equipo de desarrollo

Investigación y preparación (IV)

En la fase de **Preproducción**, se enumeran los objetivos del juego, mecánicas, retos,...

El objetivo es hacer un juego divertido mediante la integración de ideas en el diseño y luego desarrollar esas ideas utilizando conceptos artísticos y maquetas. Los hitos dentro de la fase de preproducción incluyen:

- Creación del concepto (CD). Selección y generación de ideas
- Creación de prototipos para probar las ideas que se discuten durante las sesiones de generación de ideas y lluvia de ideas
- Documento de diseño del juego (GDD). Contiene una descripción completa del juego en todos sus aspectos:
 - Descripción de los objetos, elementos y personajes en el juego (estática)
 - Interactuación, rol y comportamiento en el juego (dinámica)

Diseño (I)

Claros objetivos, tópicos y escenarios de juego

Objetivo primario de esta fase es crear las líneas maestras de 3 estructuras:

E/S

Forma comunicación/interacción HCI

Juego

Arquitectura interna del juego. Retos a superar. Relaciones causales

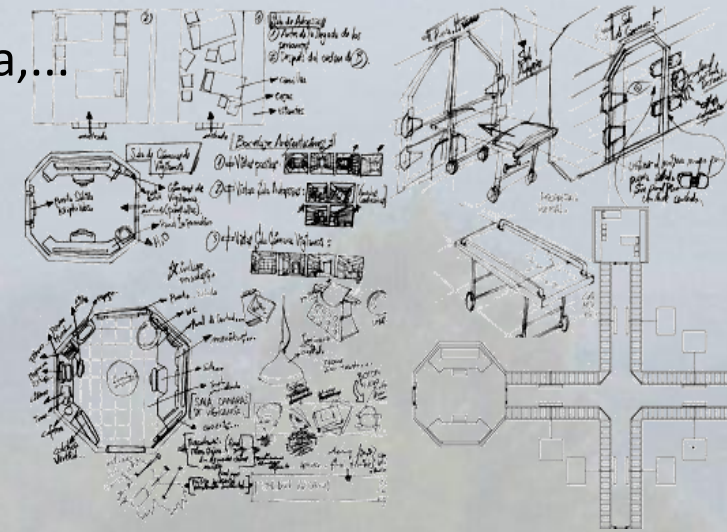
Programa

Módulos, rutinas, modelo de datos, arquitectura informática,...

Las tres estructuras se diseñan al unísono

Decisiones en una parte afecta al resto

Ciclo de coordinación entre estructuras



Diseño (II). E/S (I)

Aspecto más restrictivo de todos

La parte más importante y difícil de realizar de todo el juego

Lo que el jugador percibe del juego

Contenidos, emociones, ideas,... se canalizan a través de la E/S

Es independiente para la Entrada y la Salida

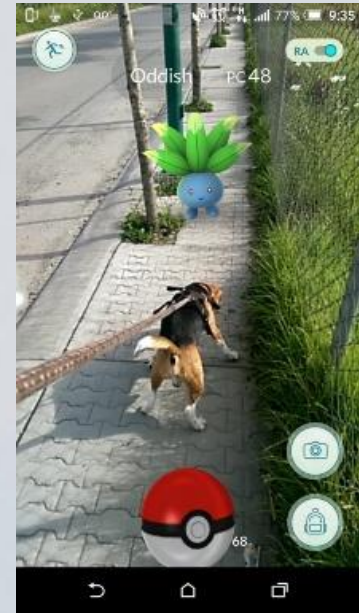
Canales no simétricos: cantidad de información, velocidad y medios

Salida: fundamentalmente gráfica. 2º plano audio y, finalmente, hápticos (vibraciones)

Entrada: principalmente

Convencionales: mandos (botones) y joysticks

No Convencionales: Actualmente hápticos (acelerómetros), geolocalización,...



Diseño (III). E/S (II)

Si opciones de juego son

Limitadas. Juego *puede* ser aburrido o perder interés

Amplias. Complejo. Disuasorio para el jugador

Determinar un interfaz es delicado. Muchas pruebas y errores

¿Riqueza o claridad?

¿Cómo controlar una central nuclear con un único joystick?

Complejidad de periférico determina las estructuras de E/S

Selección de dispositivo de entrada = Decisión de diseño crucial

Seleccionar un único tipo y ceñirse a él en exclusividad

Preferiblemente no teclado



Diseño (IV). E/S (III)

Malos diseños

Todo al servicio de necesidades de comunicación

Gráficos o sonidos espectaculares

No mejoran videojuego aburrido

Tienen sentido si mejoran la comprensión o inmersión del juego. Ayudan al jugador a estar en *flujo*

No emplear story-boards para diseñar juegos

Herramientas lineales/secuenciales

Pensadas para productos no interactivos

No soportan bifurcaciones arborescentes

Resultado = videojuego fílmico



Diseño (V). E/S (IV)

Interfaz (I)

Conjunto de dispositivos empleados en la comunicación entre el hombre y la máquina (HCI)

Pueden ser

Físicos. Hw

Virtuales. Sw

Interacción fundamentalmente

Visual. GUI. Empleo de perspectivas
1ª persona. Plano subjetivo. FPS

Al hombro (On The Shoulder – OTS)

Cenital, lateral o isométrico

Auditiva. 3 bandas: musical, efectos y diálogos

Háptica. Vibradores, subwofer, wiimote, acelerómetros,...



Diseño (VI). E/S (V)

Interfaz (II)

HUD (Head-Up Display)

OSD (On Screen Display)

Visible todo el tiempo durante el juego

Muestra información auxiliar no obligatoriamente gráfica pero necesaria

Ejemplos

Puntuaciones

Niveles de los recursos

Mini Mapas

Chat

Alertas

Nivel de juego

OSD



HUD



OSD



¿?



Diseño (VII). E/S (VI)

Interfaz (III)

Mapa de Teclas o tabla de control

Diagrama que muestra las entradas de control, acción asociada y contexto en el que se produce la acción

Action	Control	Context
Left	←	<i>all</i>
Right	→	<i>all</i>
Forward	↑	<i>all</i>
Backward	↓	<i>all</i>
Sprint	X	<i>all</i>
Pass	○	<i>Offense</i>
Lob	▲	<i>Offense</i>
Shoot	□	<i>Offense</i>
Steal	○	<i>Defense</i>
Block	▲	<i>Defense</i>
Hit	□	<i>Defense</i>

Diseño (VIII). Juego (I)

Generar un sistema concreto que muestre las fantasías del diseñador

Primer paso = Extraer los elementos claves del tópico

Elementos Claves (EC)

Actor principal en el tema del juego

Representativo del tema

Manipulable, comprensible y cuantificable

Construir juego alrededor de elementos básicos

Ejemplos:

Movimiento en juego de estrategia bélico

Afinidad personal en juego social



¿Elemento básico SIM City? <https://www.youtube.com/watch?v=SylRsLoWTgA>

La recalificación de terrenos

Diseño (IX). Juego (II)

Manipulabilidad del EC (I)

De vital importancia para el éxito del juego

Realizada a través de reglas

Restricciones mínimas y justificadas

Jugador perciba máxima libertad

Expresiva

Permitir expresar al jugador sus intenciones, estética, deseos,...

Permitirle realizar lo que quiera → Experimentar la fantasía del juego a su gusto (discrepancia modelo mental jugador vs diseñador)

Se pueden emplear varios EC. Mejor pocos y uno predominante

Cuanto más EC existan, más simple debe ser su manipulación



Diseño (X). Juego (III)

Manipulabilidad del EC (II)

Concisa

Significativa. Acción-Reacción

Focalizada: Opciones de manipulación del EC fuertemente relacionadas

CAUTION: Muchas opciones de menú o empleo de teclado denuncian EC no focalizados



Diseño (V). Conclusiones

Objetivos del diseño de

E/S: Superar restricciones

Juego: Detectar posibilidades

E/S determina limitaciones en el juego

Complejidad algorítmica visible al jugador = $f(\text{consecuencias en el juego})$

Homogenidad en nivel de especificación de detalles para todos los elementos de juego en todos sus niveles

Durante desarrollo del juego aparecen nuevas posibilidades:

Determinar si se ajustan al interfaz de juego

Se integran bien en el interfaz preexistente

Hay que repensar de nuevo el interfaz

Complica su manejo,...

Evaluación del diseño (I)

¿Diseño satisface mis objetivos?

¿Experimentará al jugador lo que yo quiero que experimente?

Verificar que dinámica de juego no

Disparará parámetros fuera de control. Juego llega a estados inestables

Impedirá un juego limpio: cantidad máxima de armamento que puede llevar un soldado, dinero acumulable en la cuenta, nº vidas ilimitado o que sea tan grande que lleve a falta de motivación,...

Evaluación del diseño (II)

Si dinámica de juego lleva a un abuso, imponer limitaciones por programa. Saturación de parámetros

Eliminar posibles atajos para obtener objetivos mediante bloqueos razonables y comprensibles

Inducir al jugador a experimentar los procesos establecidos por el diseñador

¡CUIDADO!. Evitar sensación de “pastoreo” del jugador por el diseñador



Evaluación del diseño (III)

Toma de decisión de producción del juego

De cada 100 ideas de juego, alrededor de un tercio merecen ser diseñadas. Sólo el 5% llega a ser un juego real

Abortar desarrollo de juego

- Evitará muchas pérdidas

- Permite comenzar con el análisis de otro juego

Debe meditar-se seriamente

- Si no resulta excitante

- Si éxito no se percibe

- Capacidad para desarrollarlo es insuficiente: dinero, experiencia de programación, cantidad de programadores, otros proyectos en marcha, tiempo de implementación,...

- Si marketing detecta que el esfuerzo invertido no compensa posibles beneficios → Pasarlo a casual-game y sondear reacción del mercado

Pre Programación

Tomada la decisión de producir el juego

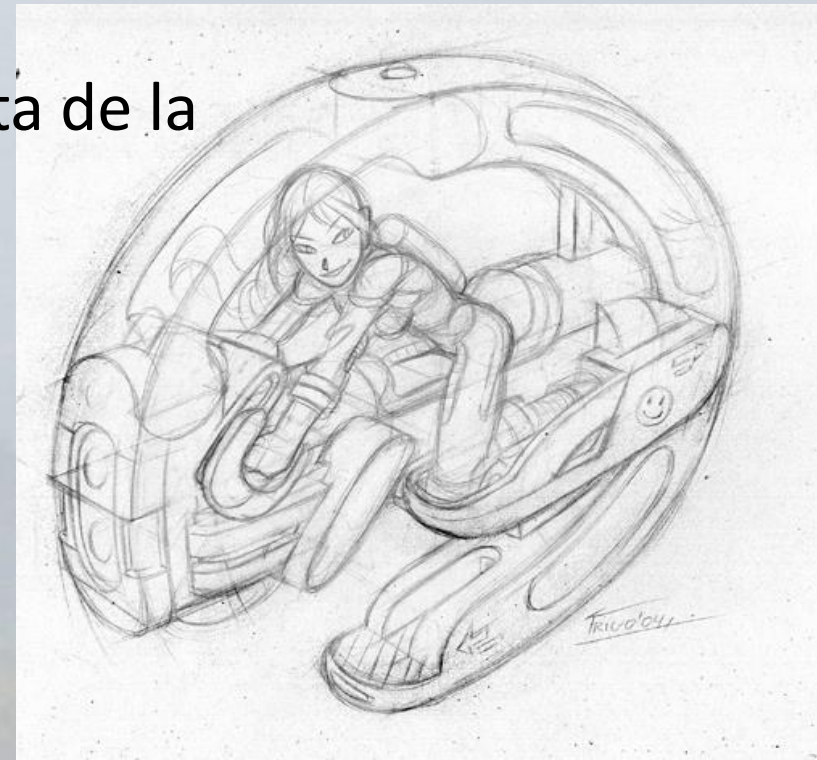
- Documentarlo todo

- Transmitir la idea a todo el equipo con la mayor fidelidad posible

- Transcribir borradores, anotaciones, ideas, hacer bocetos más concisos

- Enfatizar descripción desde punto de vista de la experiencia del jugador

- No tomar decisiones técnicas



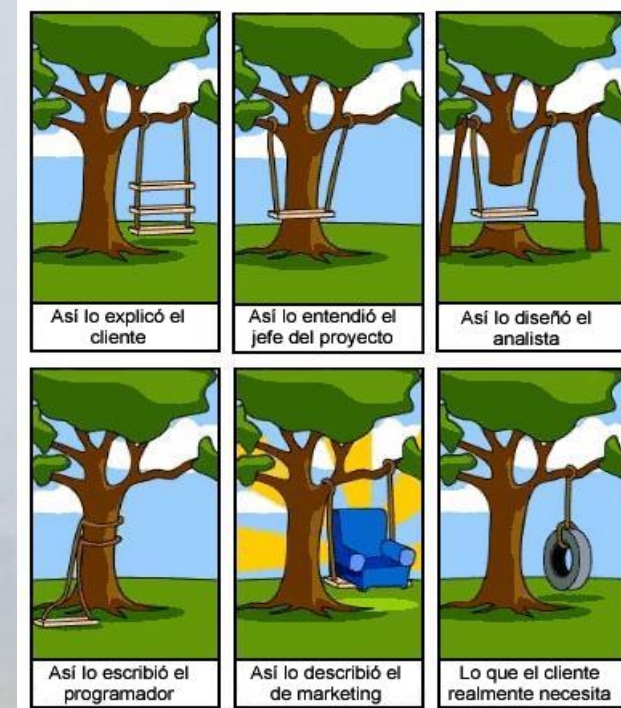
Programación (I)

Fase tediosa y larga

La más costosa económicamente

Difícilmente producción del juego fracasa por falta de talento

Éxito del juego no se debe fundamentalmente a una programación buena o espectacular



Programación (II)

Convierte la parte de E/S y la estructura del juego en un producto real

Reutilizar código genérico realizado para otros programas: APIs, grafos de escena, código propietario,...

Priorizar las funciones fundamentales

CUIDADO: funciones secundarias pueden eliminar recursos para las fundamentales: memoria, CPU,...

Tendencia a la programación paralela GPUs/CPUs, multithreading,...

Documentación interna y externa

Empleo de tecnologías y metodologías de Ing. SW

```
#define NOTA_NO_VALIDA 10
#define MATRIZ_MAXIMOS 10
#define PRIMER_ALUMNO 0
void Imprimir(float Notas [TOTAL_ALUMNOS])
{
    unsigned int Alumno = PRIMER_ALUMNO;
    while (Alumno < TOTAL_ALUMNOS)
    {
        printf("La nota del alumno %d es %f", Alumno, Notas[Alumno++]);
    }
}

/*Iniciación de índice del alumno*/
void Introducir(float Notas [TOTAL_ALUMNOS])
{
    unsigned int Alumno = PRIMER_ALUMNO;
    while (Alumno < TOTAL_ALUMNOS)
    {
        printf("Por favor, introduzca la nota a almacenar");
        scanf("%f", &Notas[Alumno]);
        Alumno++;
    }
}

/*Iniciación de índice del alumno*/
char Estadisticas(float Notas [TOTAL_ALUMNOS])
{
    unsigned int Alumno = PRIMER_ALUMNO;
    while (Alumno < TOTAL_ALUMNOS)
    {
        printf("La nota del alumno %d es %f", Alumno, Notas[Alumno++]);
    }
}
```


Programación (III)

Pre-Alfa

Actividades realizadas durante el proyecto de software antes de las pruebas formales: análisis de requisitos, diseño de software, desarrollo de software y pruebas

Se emplean **hitos** que incluyen conjuntos específicos de funciones que se conquistan a etapas pequeñas. Tan pronto como se completa una funcionalidad, se pasa al siguiente hito

Prueba inicial del juego

Permite la retroalimentación de los usuarios, solucionar problemas, probar características y saber qué gustaron exactamente los jugadores



Programación (IV)

Alfa

Es la primera versión completa del programa, la cual es enviada a los verificadores para probarla

El producto todavía es inestable

Todavía presenta algunos errores

Funcionalidad incompleta, pero satisface la mayoría de los requisitos



Programación (V)



Beta

La primera versión completa de un videojuego

Es MUY posible que sea inestable

Versión preliminar

Etapla comienza cuando los desarrolladores congelan las características del videojuego

No serán agregadas más características a esta versión

Solamente se harán pequeñas ediciones o se corregirán errores

Paso intermedio en el ciclo de desarrollo completo

Los desarrolladores las lanzan a un grupo de probadores de betas o beta testers. A veces al público en general para una prueba de usuario

Los probadores suministran errores que encuentran y nuevas características que quisieran ver en la versión final



Programación (VI)

Versión candidata a definitiva (RC)

Versión muy estable

Implementa todas las funciones del diseño

Prácticamente libre de errores

Videojuego preparado para publicarse como versión definitiva para distribución comercial a usuarios finales

Algunas empresas de desarrollo utilizan siguientes letras alfabeto griego (*gamma*, *delta*,...) para versiones que están prácticamente completas pero todavía en pruebas

También se suele usar RC+nº indicando subversiones cada vez más cercanas al producto final

Versión *omega* para versiones que se creen libres de errores y se hallan en el proceso final de pruebas



Programación (VI)

Versión de disponibilidad general

También llamada *versión dorada*

Normalmente es casi idéntica a la versión candidata final, con sólo correcciones de última hora

El videojuego ha sido completado y que está siendo producido masivamente y estará en venta próximamente

El término *dorado* se refiere al uso del disco maestro de los discos musicales de vinilo que se estampaban en oro y que se usaba para enviar la versión final a los fabricantes



Testeo (I)

Las pruebas del juego garantizan la calidad y la diversión

Estas pruebas encontrarán errores

La recepción de pruebas darán al patrocinador la confianza el videojuego antes del lanzamiento final

Los comentarios recibidos durante la prueba beta abierta son importantes para el éxito del juego



Testeo (II)

Los hitos de esta fase consisten en:

Open Beta. Realización de las pruebas más inclusivas

Poner a prueba la dificultad, la recepción y la búsqueda de errores

Realizar cambios de última hora antes del lanzamiento final del juego

Prelanzamiento. Emplear a los medios para establecer una base de fans.
Generar expectativas en RR.SS.

Lanzamiento final. Día de comienzo de la distribución al público final del juego



Testeo (III)

Pule y refina el diseño del juego

Suele obligar a modificar código y excepcionalmente el diseño de E/S

Modificaciones asumibles: demasiadas piezas de juego simultáneamente, juego aburrido, falta acción, colorido o iluminación insuficiente,...

Modificaciones fatales: conflicto severo entre dos elementos de juego importantes. Solución parchear o abortar el desarrollo

Acumulación de parches → proyecto inviable → abortar



Testeo (IV)

Problemas encontrados pueden deberse a malos diseños

No parchear ni ir a soluciones fáciles e inmediatas

Analizar razón íntima del fallo

Depurar la raíz, no el efecto

Seleccionar del abanico de soluciones aquella que refuerza consecución de objetivos del juego

Dos tipos

Interno. Elimina errores de programación y de juego

Externo. No errores internos. Beta testers. Realizado por jugadores no involucrados en el desarrollo del juego. Descubrir fallos en la estructura del juego, recomendar modificaciones y mejoras, fallos visuales,...



Testeo (V)

Beta tester (I)

Seleccionados con mucho cuidado

Clausula de confidencialidad

No más de media docena. Cuidado: sesgo

Jugón. Experiencia de juego amplia

Preferiblemente diseñadores de otros juegos

Conocer jugador: preferencias, personalidad,... Corregir sesgo

Testeo en grupos amplios de gente real y amiguetes

Genera información pobre: críticas vagas, no concreción en cambios de diseño,...

Pocas conclusiones prácticas. Incompetencia informática



Testeo (VI)

Beta tester (II)

Deben leerse un manual preliminar

Contenido del manual suficiente para evitar críticas al juego gratuitas

No acompañar al testeador durante las pruebas.

Contaminación. Invalidación de resultados

Evitar reuniones prematuras. Dejar pasar al menos una semana de juego

Entrevista igual para todos. Puede emplearse 3ª persona.

Mejora asepsia contestaciones. Limita interacción directa e improvisación de otras preguntas, aclaraciones,...



Testeo (VII)

Beta tester (III)

90% respuestas de testers se descartan:

Incompatibles con objetivos del juego

Inalcanzables con los recursos disponibles:
presupuestos, recursos HW disponibles,...

Requieren modificaciones importantes del juego.

Mejor para la segunda versión o para otro juego



Pulido

Se desarrolla en paralelo a las fases de testeo y de desarrollo final del juego

Resistir urgencia de comercialización

CONGELAR ESPECIFICACIONES

No yaques, ni porquenos

Ajustes finos, pequeños errores, realimentación de testeadores,...

Siempre aparecerán nuevos errores no detectados

Cuando se publica, ya no hay vuelta atrás

Finalizar y cuidar el diseño del manual



Manual

Desarrolla toda la fantasía del juego

Imágenes sugerentes

Metáfora en toda su extensión

Cubre limitaciones del computador

Genera la atmósfera del juego e involucra y sumerge al jugador en el mundo de fantasía

Clarifica conceptos y mejora el rendimiento del juego y la satisfacción del jugador

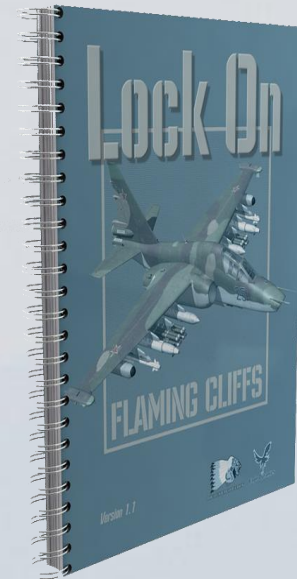
Comenzar a redactar el manual al mismo tiempo que el código →

Clarifica conceptos y evita errores

Realimentación entre código y manual

Canal de comunicación entre desarrolladores y diseñadores

Primera versión por diseñador. Después por escritor profesional



Push(I)



Etapa que permite a los usuarios con contenido premium

- Mejorar el juego con actualizaciones de postproducción

- Mantener el juego en ventas continuadas

- Dejar que los jugadores agreguen lo que quieran agregar al juego

Esta etapa es más como una maratón y su éxito o fracaso influirá el beneficio final obtenido

Los hitos dentro de la estrategia PUSH consisten en:

- Operaciones y mantenimiento. Pueden aparecer mejoras o errores que exigen mantenimiento y reparaciones. Típicamente se ofrecen actualizaciones

- Contenido Premium. Permite ampliar la funcionalidad del juego o permite a los jugadores agregar lo que quieran por una pequeña tarifa

Métricas de éxito del juego: medidas en descargas totales, revisiones positivas e ingresos totales generados en la primera semana, mes o año

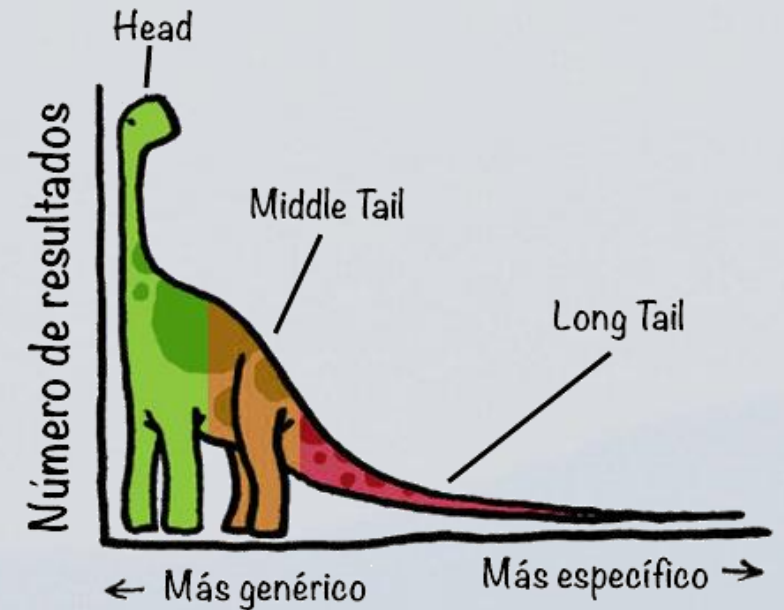
Push(II)

Long Tail (I)

Expresión popularizada por Chris Anderson para describir tipos de negocios y modelos económicos basados en las TICs tales como Amazon o Netflix

Se utiliza normalmente en estadística en relación con distribuciones de riqueza o con el uso del vocabulario: Ley de potencias, [distribuciones de Pareto](#) y [distribuciones de Lévy](#)

Es un modelo de negocio tradicional se enseñaba que los productos que se deben vender son los que tienen mayor rotación



Push(III)

Long Tail (II)

Internet y el entorno digital han cambiado las reglas del mercado reduciendo el coste de almacenamiento y distribución

Ya no es necesario focalizar el negocio en pocos productos. Ahora existen dos mercados:

El mercado de masas: centrado en el alto rendimiento de pocos productos

Los nichos de mercados: se basa en la suma o acumulación de todas las pequeñas ventas de muchos productos, que pueden igualar o superar al anterior



Push(IV)

Long Tail (III). Mercado de masas

Mercado tradicional físico

Limitaciones geográficas y físicas:

- Se necesita concentrar una audiencia numerosa y generar una gran demanda

- Ámbito geográfico manejable y rentable

- El coste de almacenaje y de distribución provoca que se centren en pocos artículos de fácil y rápida venta.

Ejemplos

Tiendas de discos y librerías físicas. Sólo venden discos de interés para una audiencia amplia. Temas promocionados en radios comerciales o best sellers.

Alto volumen de ventas de productos de moda

Regla del 80/20 recomienda centrarse en el 20% de los productos para obtener el 80% de ingresos



Push(V)

Long Tail (IV). Nicho de mercados

Entorno digital. Costes de almacenaje y distribución de productos digitales es nulo. Nueva economía sin fricción

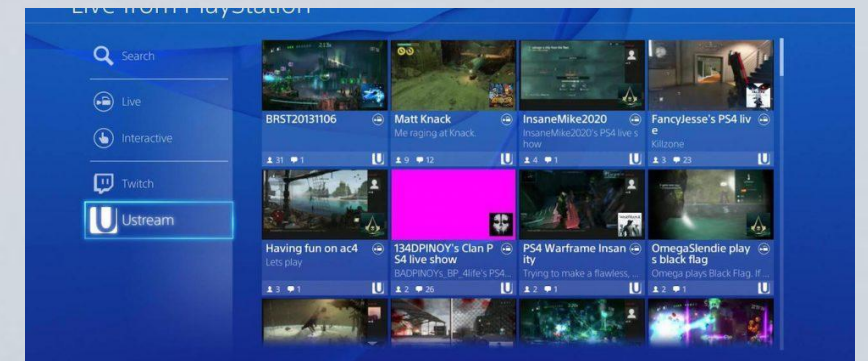
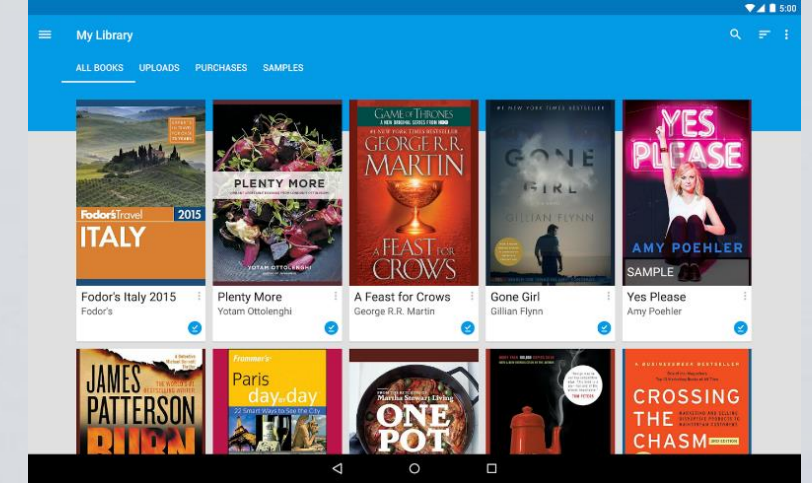
Amplio catálogo de productos y cubrir una extensa gama de gustos e intereses

Ejemplos

Amazon, ebay, PlayStation Network, Xbox Live, Play Store, Apple App Store, Steam, Netflix, HBO,...

Venden pocas unidades de cada producto, pero sumados todos los productos pueden generar un gran beneficio

Ofrecen productos muy populares, menos populares y “raros”, descatalogados, antiguos, frikies,...



Post Mortem

Realizar un rastreo del juego: foros, webs especializadas, prensa, TV,...

Muchas críticas provienen de personas

Poco expertas

No se han leído el manual

Pirateado el programa y no les funciona bien o no han tenido acceso al manual

Juegan de oídas o sienten envidias

Prestar atención a justificación de contenidos

A veces juegos se denigran sólo por sus objetivos

Éxito: se venda mucho. Genera ingresos para generar más juegos y mensaje llega a más gente



Bibliografía

Cap. 5. The Game Design Sequence. The Art of Computer Game Design. Chris Crawford. <http://library.vancouver.wsu.edu/art-computer-game-design>

The Game Production Pipeline: Concept to Completion. IGN. 15/03/2006

Cap. 2. Fundamentals of Game Design II. Ernest Adams, New Riders

Cap 8.4. SOFTWARE ENGINEERING, 9th. Edition, Ian Sommerville, ISBN 978-0-13-703515-1

Documentación generada por
Dr. Ramón Mollá Vayá
Sección de Informática Gráfica
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia

Reconocimiento-NoComercial-CompartirIgual 2.5

Usted es libre de:

copiar, distribuir y comunicar públicamente la obra
hacer obras derivadas bajo las condiciones siguientes:



Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador.



No comercial. No puede utilizar esta obra para fines comerciales.



Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor

Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.