

Arquitectura de videojuegos

Máster en Programación de videojuegos

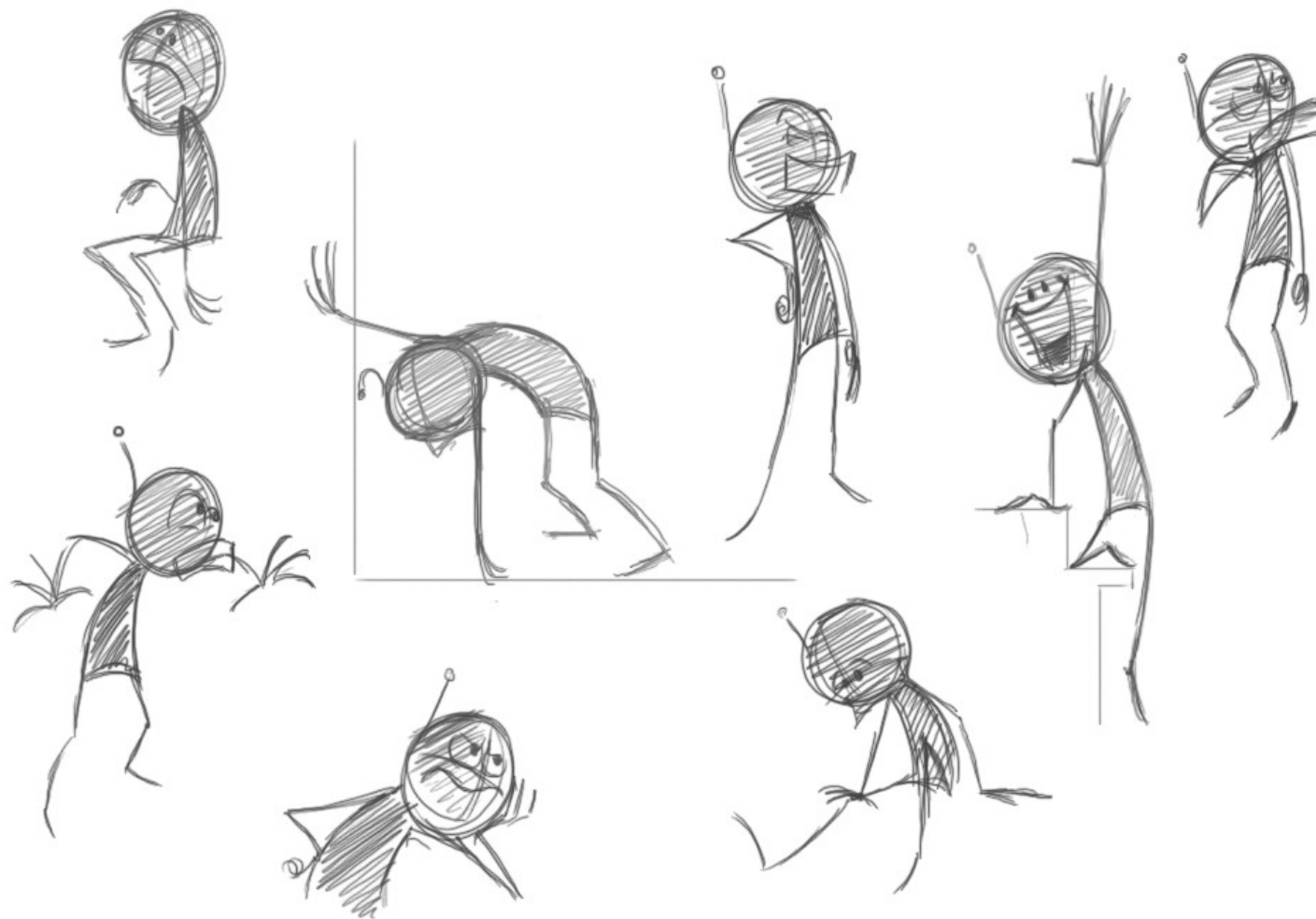


Javier Arévalo

2º Trimestre
Curso 2013-2014

Arquitectura de videojuegos

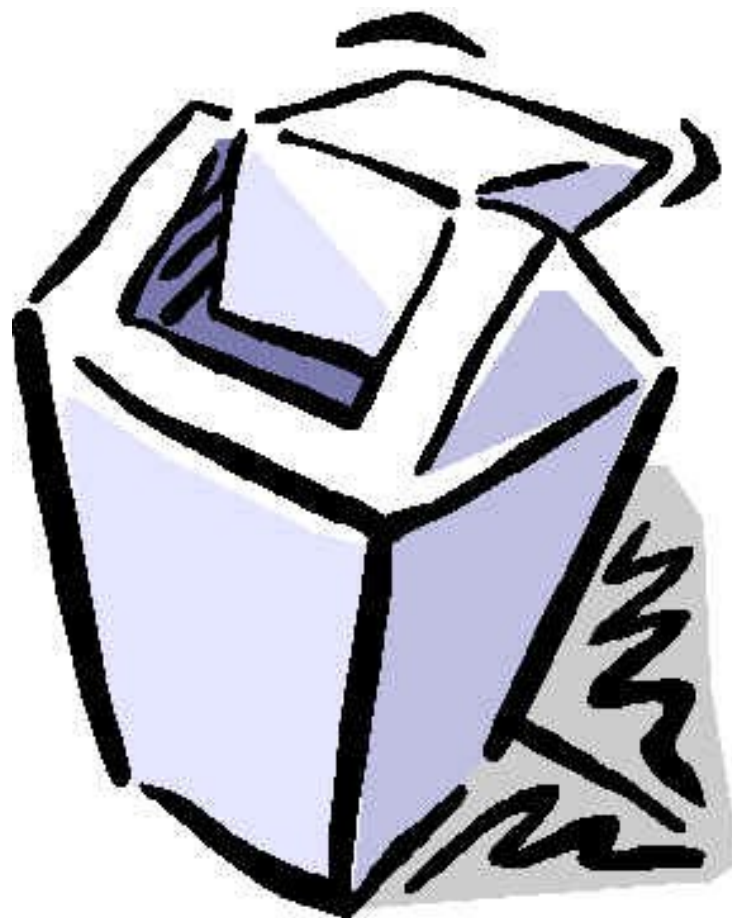
Prototipado



Prototipo

- Una versión limitada del juego o aplicación
- Centrada en una o dos funcionalidades concretas
- Objetivo: contestar a una pregunta o preguntas concretas
- Separado del desarrollo principal
- Tras desarrollarse y usarse, se tira TODO menos el conocimiento adquirido

Un momento...



Un momento...

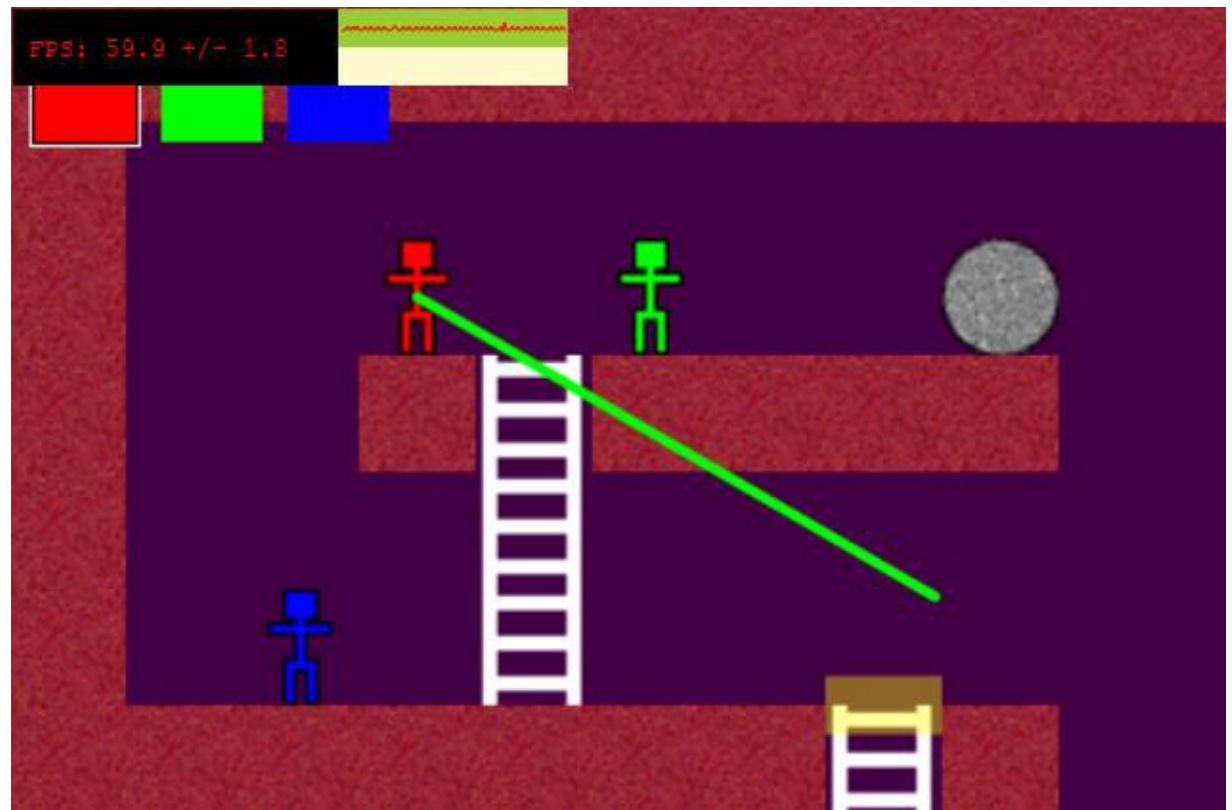
- Tirarlo?? Pues si!
- La filosofia de desarrollo y la arquitectura de un prototipo es muy diferente que la de un juego o aplicación completos y publicables
- Seria un grave error continuar el juego con una base endeble





Version limitada

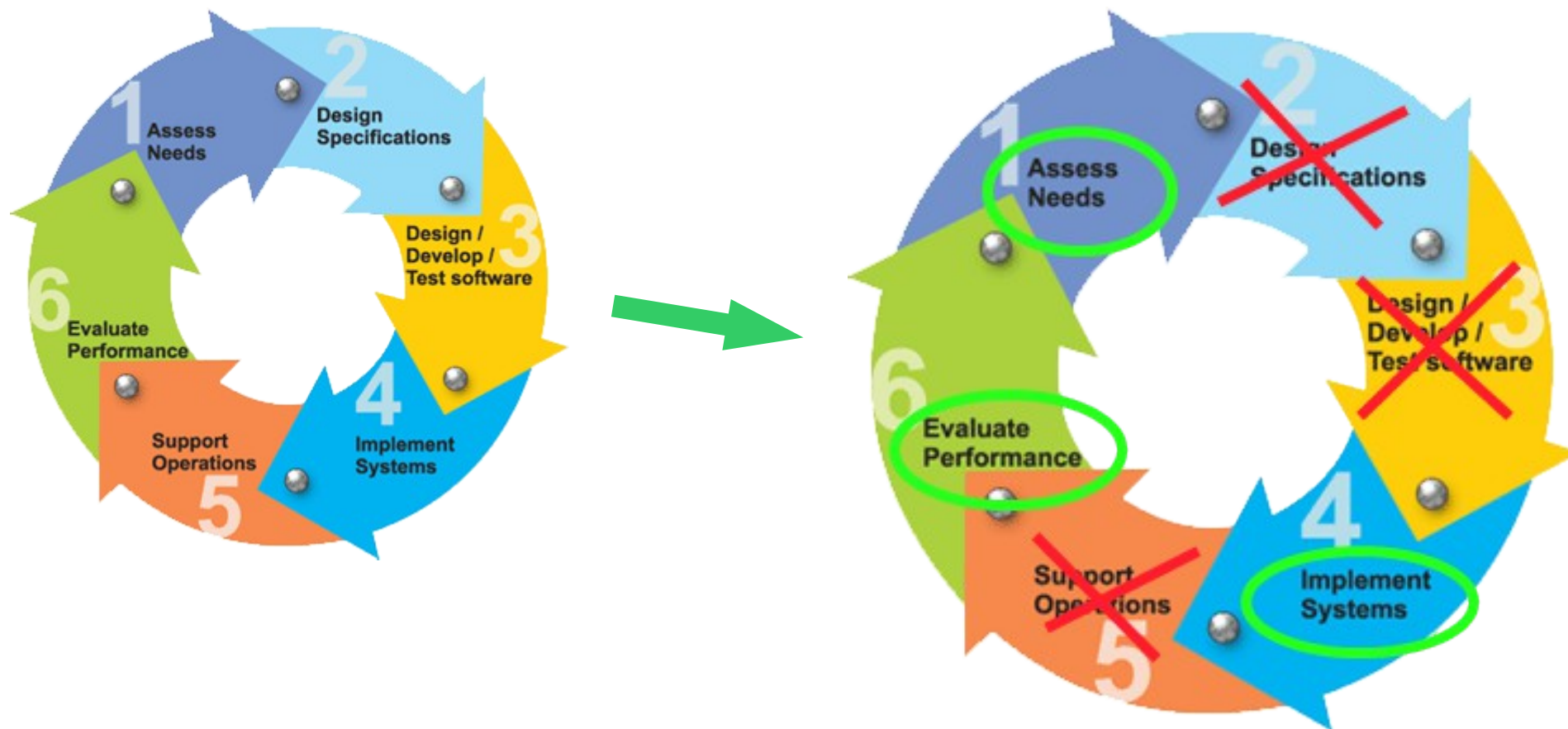
- Graficos basicos y puramente funcionales
 - A no ser que el prototipo sea de una tecnologia grafica
 - Coder art!



Version limitada

- Lo mismo con todo lo demas:
 - No hacer 5 variantes del mismo enemigo
 - No preocuparse de equilibrar la progresion
 - No poner sonidos o musica
 - No hacer menus y pantallas de settings
 - Escenarios pequeños con solo lo imprescindible
 - Nada de 'atrezzo', efectos especiales, etc
- A no ser que ESO sea lo que quieres prototipar

Ingenieria... a medias



Ingenieria

- El objetivo es desarrollar rápido, e iterar aun mas rápido, sobre las preguntas que se quieren responder
- No buscamos código mantenible ni sistemas reutilizables o extensibles
- No necesitamos optimizacion ni seguridad
- Huir de la necesidad de usar datos externos
 - Archivos de definición, bases de datos, etc

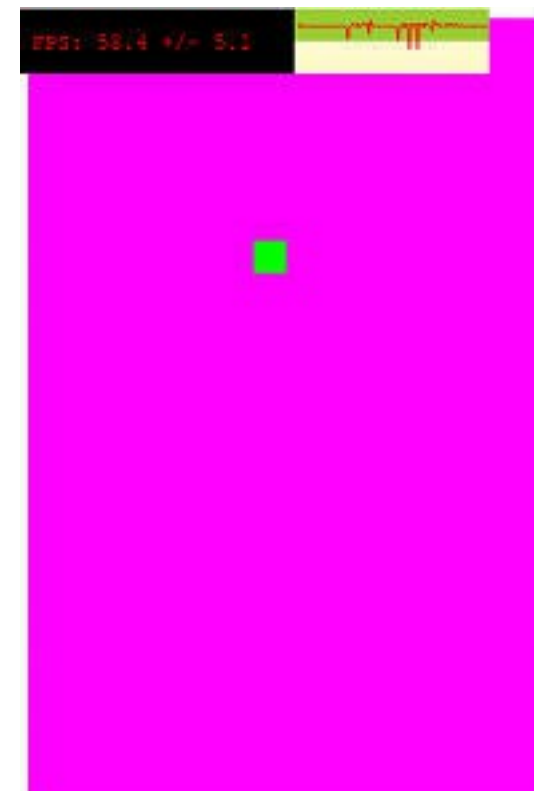
Un poco de código

```
<html><body><script>
canvas = document.createElement("canvas");
canvas.width = 320;
canvas.height = 480;
canvas.style.background = "#F0F";
context = canvas.getContext("2d");
document.body.appendChild(canvas);

playerX = 100;
playerY = 100;
keyboard = [];
window.onkeydown = function(e) {keyboard[e.keyCode]=true};
window.onkeyup = function(e) {keyboard[e.keyCode]=false};

setInterval(function() {
    if (keyboard[37]) playerX--;
    if (keyboard[39]) playerX++;
    if (keyboard[38]) playerY--;
    if (keyboard[40]) playerY++;

    context.clearRect(0, 0, canvas.width, canvas.height);
    context.fillStyle = "#0F0";
    context.fillRect(playerX, playerY, 20, 20);
}, 16);
</script></body></html>
```



Un poco de código

- Variables globales (playerX, canvas, etc)
- Números mágicos (#F0F, 37, 20, etc)
- Funciones, las justas
- Creando las cosas según las necesitamos
- El código es PEQUEÑO y dúctil
- Hasta cuanto podemos extender este modelo?
 - Hasta que deje de ser cómodo para modificar

Un poco de código

```
function initplayer() {
    playerX = 0;
    playerY = 0;
}

keyboard = [];

function updatekey(keycode, ispressed) {
    keyboard[keycode] = ispressed;
}

function moveplayer() {
    if (keyboard[KEY_LEFT_ARROW])    playerX--;
    if (keyboard[KEY_RIGHT_ARROW])   playerX++;
    if (keyboard[KEY_UP_ARROW])      playerY--;
    if (keyboard[KEY_DOWN_ARROW])    playerY++;
}

function draw() {
    context.clearRect(0, 0, canvas.width, canvas.height);
    context.fillStyle = "white";
    context.fillRect(playerX, playerY, 20, 100);
}
```



DO NOT SHAVE

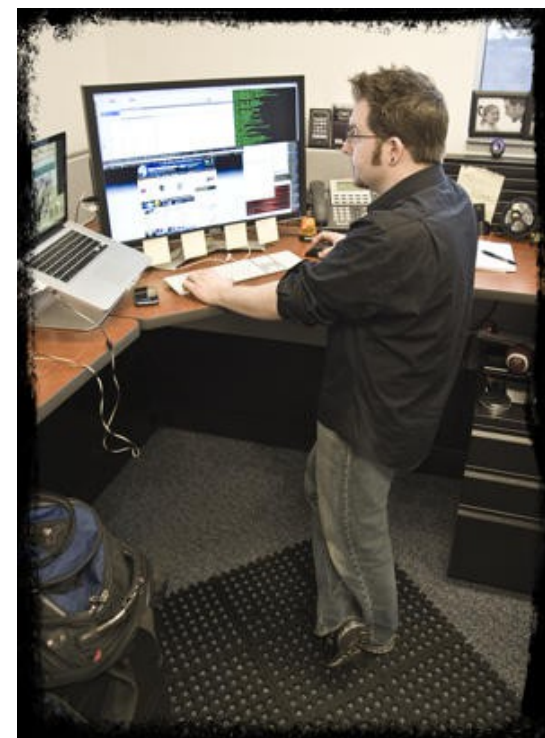
A HAIRY YAK IS A HAPPY YAK

Datos

```
var mapDef = [
  '#####',
  '#',
  '# 1 2 O',
  '# #H### V',
  '# H',
  '# 3 H',
  '#####H# ###H#####',
  '# H H',
  '# H A H',
  '# #####H## H',
  '# H H',
  '# H H',
  '#####H#####',
  '# H',
  '#####',
  ];
```

- Imbuidos en código
- Procesados en run-time
- Poco flexibles para usos futuros
- Pero sencillos e inmediatos de editar

Comodidad



Características habituales

- Sin menú principal
 - Se entra directamente al juego
 - Como mucho con un selector simple de nivel
- Sin metajuego
 - Leaderboards, level cap, achievements, etc
 - Hace falta tener juego para tener metajuego
- Sin final de partida
 - Excepto que se quiera prototipar algo ahí

Tecnologías

- Unreal Engine, otros middleware
- Tecnología del juego anterior
- Código ad-hoc
- Herramientas gráficas (Photoshop, Maya, etc)
- Papel u otros materiales

Middleware

- Todo tipo de funcionalidades listas para usar
 - Graficos, menús
 - Herramientas, editores de mapa
 - Control de jugador, AI, navegacion
 - Red y multijugador
- Grandes y pesados, no ágiles
- Orientadas a un estilo
 - No agiles para prototipar un estilo nuevo

Tecnología anterior

- Experiencia solida para desarrollar sin tiempo de puesta en marcha
- Muchos contenidos ya existentes listos para reutilizar
- Mismos problemas que con middleware
 - Orientada a un estilo existente
 - Grandes y orientadas a producción: poco eficientes para desarrollo rápido

Código *ad-hoc*

- Todo el código desarrollado de cero
 - Aunque el copy-paste vale
- Muy útil para ideas completamente nuevas
 - No hay ataduras preexistentes
- Costoso si los conceptos a prototipar son muy detallados
 - P.ej. Sistema de combos para un juego de tipo Brawler
- Recomendable lenguajes dinámicos (Javascript, Python) o entornos completos (Unity)

Código *ad-hoc*

- Conviene acumular una librería de utilidades
 - Pintar rectángulos rotados con color y alpha
 - Pintar bitmaps escalados y rotados
 - Geometría, Vectores, Rectángulos, ángulos, Color
 - Lectura de ficheros, JSON, etc
 - Leer teclado, ratón, gamepad sencillo
 - Dibujar formas básicas 3D en wireframe
 - Servicios de red, framework de servidor

Herramientas gráficas

- Photoshop, Maya, etc
- Para prototipar estilos y tecnologías visuales
 - Efectos especiales
 - Shaders, iluminación
 - Sistemas de compresión de imagen
 - Animaciones
- Niveles, decorados y relaciones espaciales

Físicos

- Usar papel, LEGOs, etc
- Muy útil para construir y visualizar niveles y escenarios
 - Dimensiones
 - Navegabilidad
 - Paradojas espaciales
- “Simular” el juego moviendo fichas
 - Al estilo de juegos de tablero

Finalmente...

- Cuando las preguntas estén respondidas
 - Formular mas preguntas
 - o
 - Tirarlo
- Si puedes extender el prototipo para el juego
 - El prototipo hace demasiadas cosas
 - o
 - El juego va a tener unos cimientos endebles

Vuestro turno!

Preguntas?

