

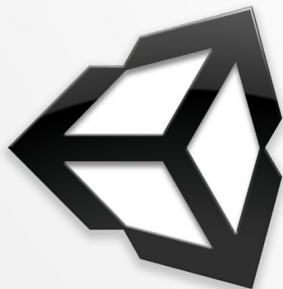
ECS

Un patrón de diseño para el desarrollo
de videojuegos

Agenda

- Introducción
- Patrones de diseño
- Fundamentos de OOP
- ECS
 - Motivación
 - Conceptos
 - Implementación
 - Ventajas y desventajas

Introducción

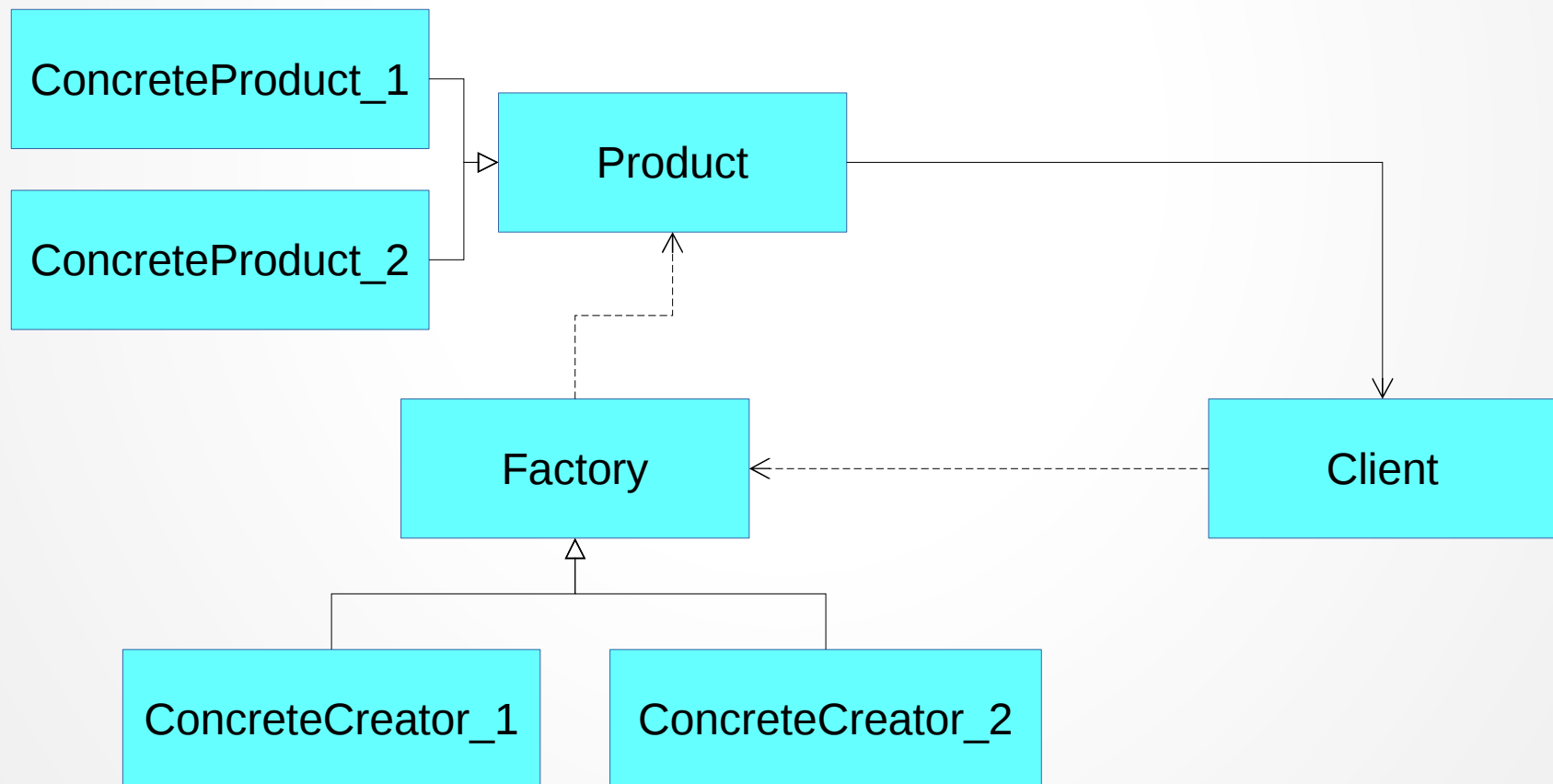


Patrones de diseño

- Soluciones generales a problemas recurrentes
 - Buenas prácticas
 - Varias categorías
 - Patrones de creación
 - Patrones arquitectónicos
 - Patrones de comportamiento
 - Patrones de concurrencia

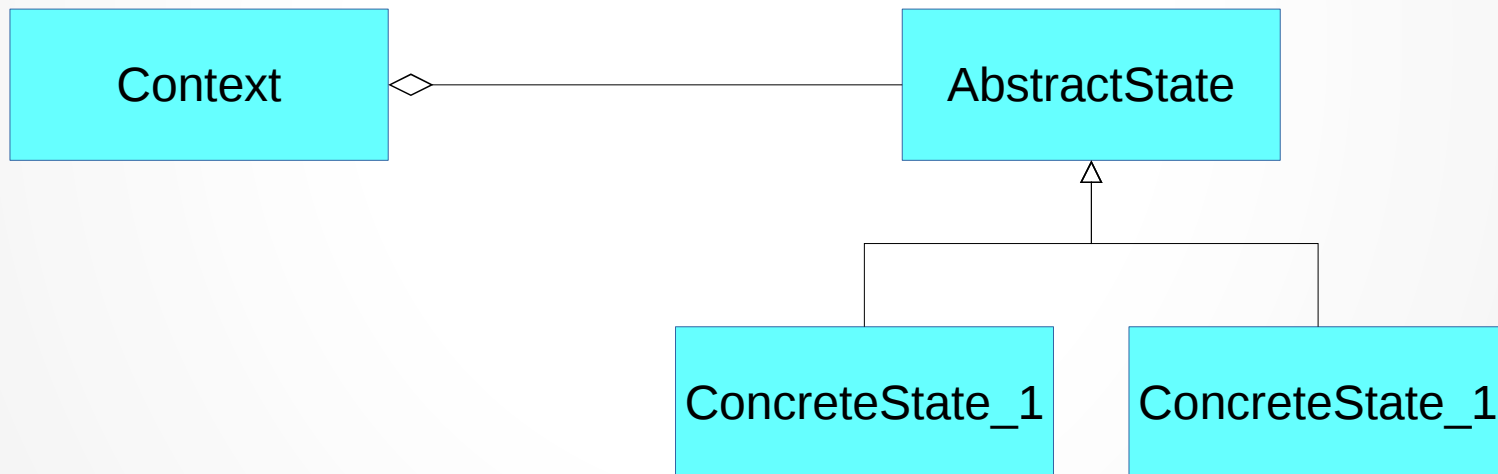
Patrones de diseño

- Ejemplo
 - *Factory method*



Patrones de diseño

- Ejemplo
 - *State Pattern*



Fundamentos de *OOP*

- Paradigma de programación
 - Conceptos
 - Clase
 - Objeto
 - Relaciones
 - Encapsulamiento
 - Herencia
 - Polimorfismo

Fundamentos de *OOP*

- Clases
 - Atributo
 - Método
- Objetos
 - Instancia
 - Estado

Carro
+Modelo:String -Fabricante:String -NoDeSerie:Int -Velocidad[2]:Float
+GirarDerecha():void +GirarIzquierda():void +Acelerar(Float):void +Frenar():void +GetNoDeSerie():Int

Fundamentos de *OOP*

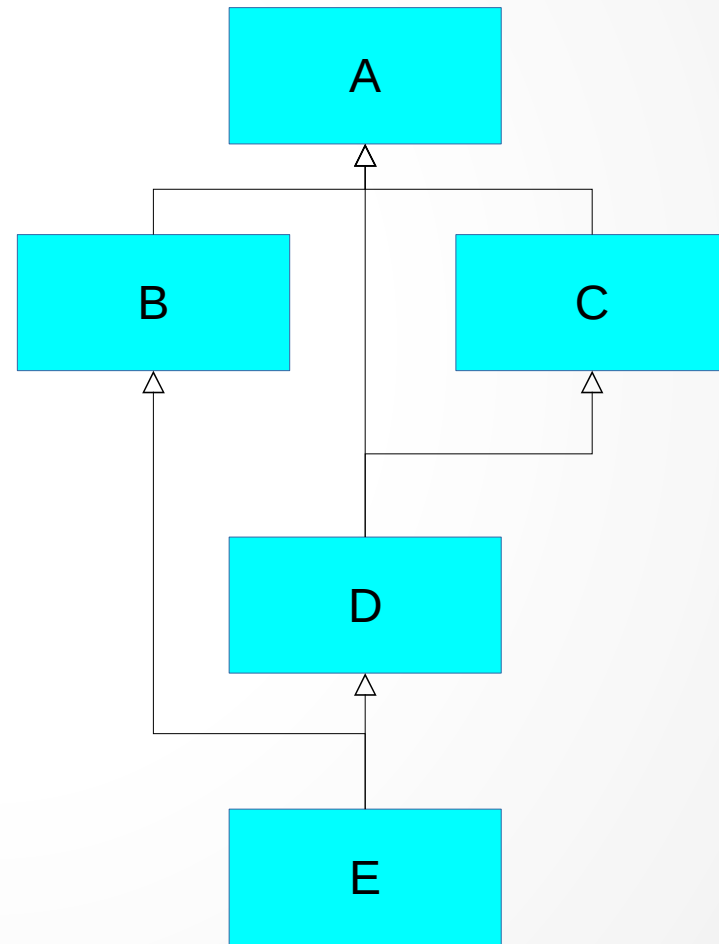
- Relaciones

- Asociación
- Agregación
- Composición
- Dependencia



Fundamentos de *OOP*

- Herencia
 - Simple
 - Múltiple
 - Métodos virtuales

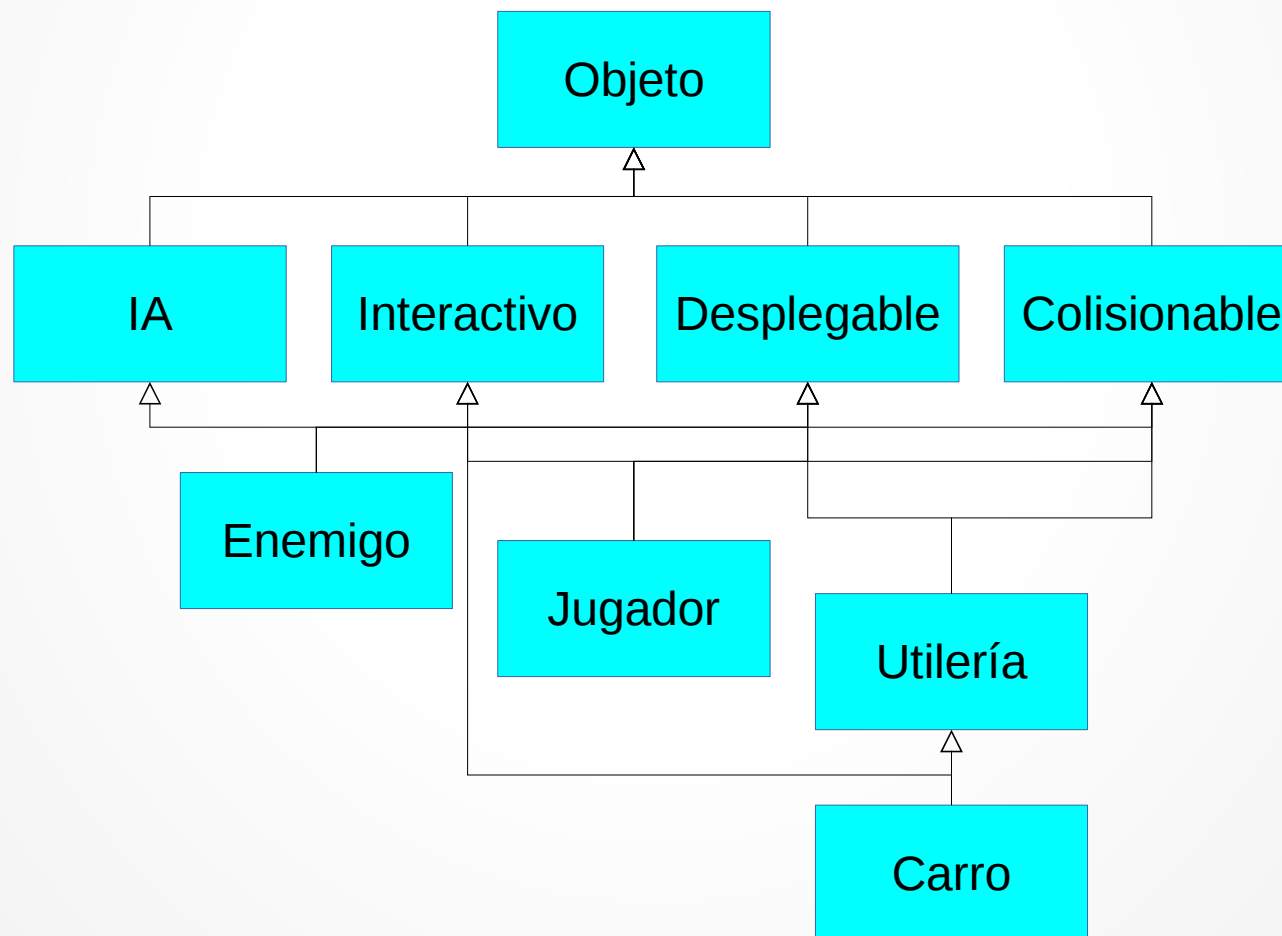


ECS

- Definición
 - Patrón de diseño arquitectónico
 - Componentes fundamentales
 - Entidades
 - Componentes
 - Sistemas
 - Programación dirigida por datos
 - Composición sobre herencia

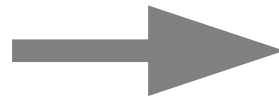
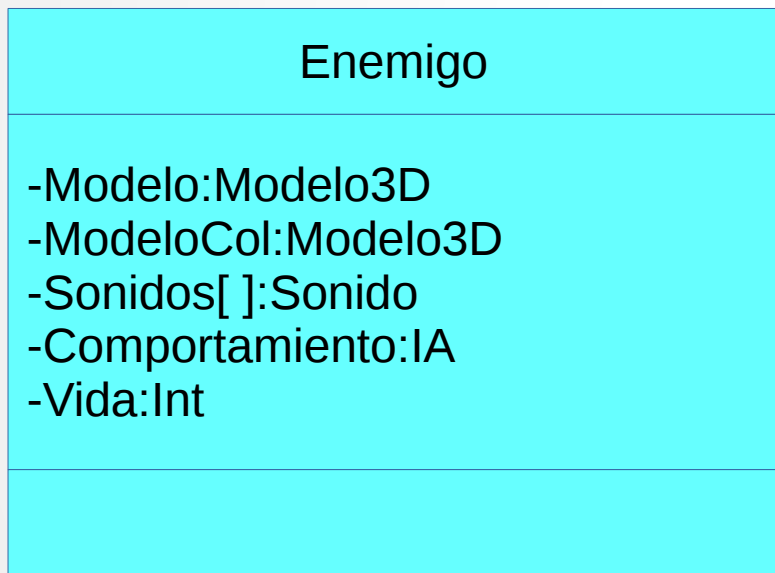
ECS: Motivación

- Evitar herencias múltiples



ECS: Motivación

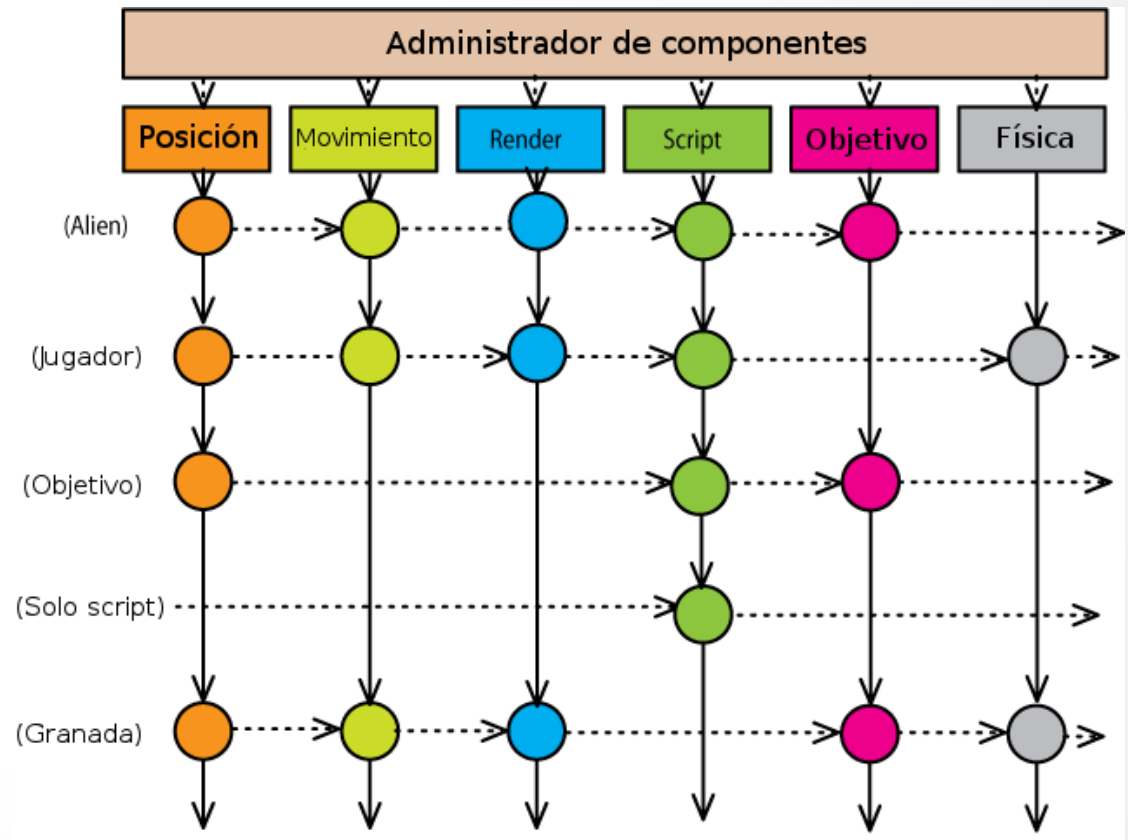
- Aislar dominios de aplicación



```
if(colisionConPiso() && (estado != INVISIBLE)) {  
    reproducirSonido(SONIDO_PISAR);  
}
```

ECS: Conceptos

- Fundamentales
 - Entidades
 - Componentes
 - Sistemas
- Adicionales
 - Aspectos
 - Grupos
 - Motores



ECS: Implementación

- Implementación de componentes
 - Estructura de datos pasiva
 - Solo atributos y/o métodos *Get* y *Set*

Posición
+PosX:Float +PosY:Float

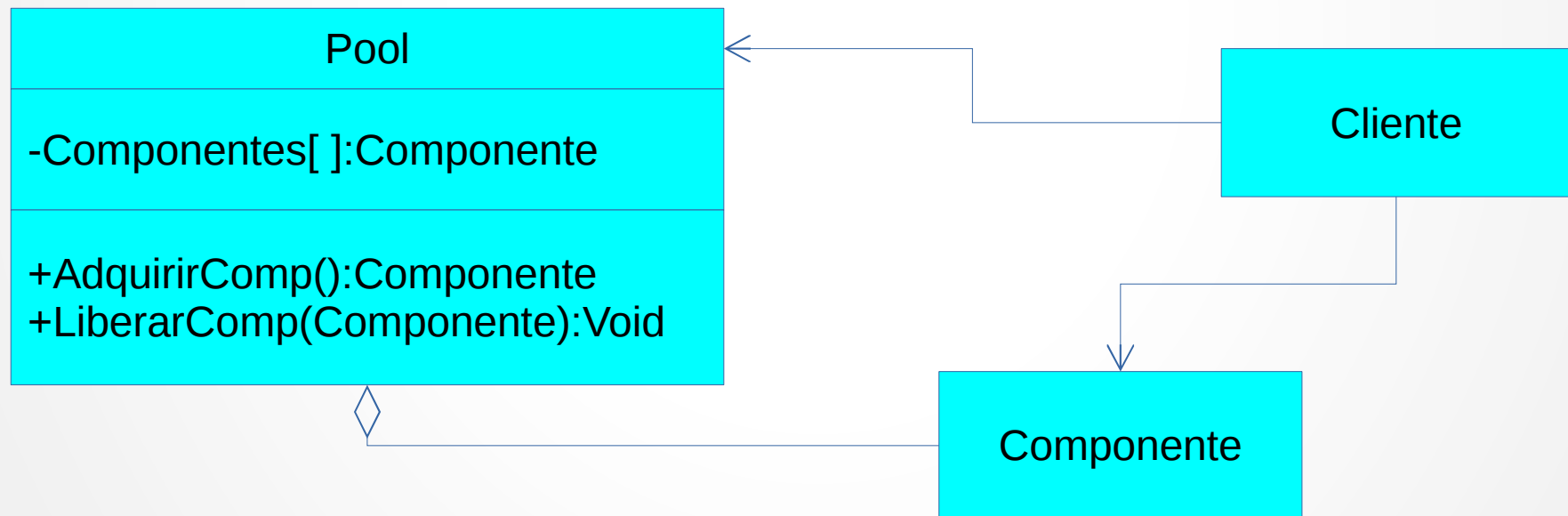
ECS: Implementación

- Implementación de entidades (2 variantes)
 - Clase entidad con una lista de componentes
 - Indice a una tabla de componentes

Entity_id	Position	Velocity	RenderModel	CollisionMode I
1	1	1	1	1
2	2	NULL	NULL	2
3	3	NULL	2	3

ECS: Implementación

- ¿Y si agregamos entidades o componentes?
 - Reutilizar índices de entidades
 - Usar *pools* de componentes



ECS: Implementación

- Implementación de sistemas
 - Pertenecer a un motor
 - Definir aspectos sobre los que trabajan al momento de ser creados
 - Procesan entidades de una en una
 - Poseen un único dominio

Sistema
+Procesar(Entidad, Float):Void

ECS: Implementación

- ¿Cuándo se procesan las entidades?
 - El motor llama a los sistemas cuando...
 - Comienza un *frame*
 - Se vence un *timer*
 - Manualmente
 - En cada llamada a un sistema se procesa una sola entidad

ECS: Implementación

- ¿Como comunicar sistemas entre si?
 - Paso de mensajes
 - *Observer pattern*
 - Componentes bandera o banderas e los componentes

ECS: Ventajas

- Código más fácil de mantener
- Composición de entidades permite cambiar comportamientos muy eficientemente
- Las tablas de entidades y sus componentes pueden guardarse en una BD relacional muy fácilmente

ECS: Desventajas

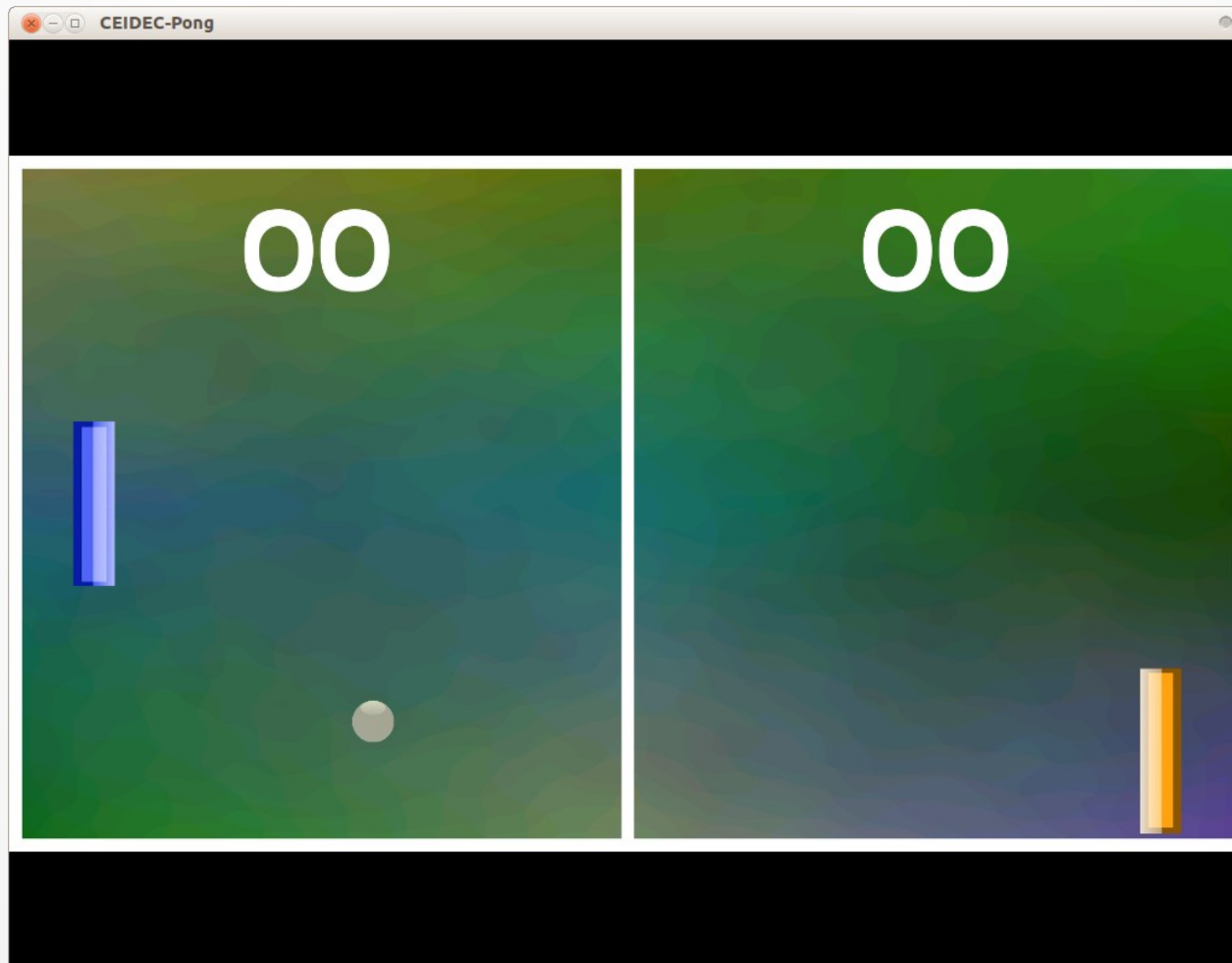
- ECS es complejo y su implementación no es trivial
- Su rendimiento depende de como se implemente la búsqueda de entidades por sus aspectos

ECS: Implementaciones

- LibGDX - Ashley
 - Java y Scala
 - Desarrollo activo
- Artemis ES Framework
 - Java, versiones extraoficiales para C++ y Python
 - Descontinuado
- Unity 3D

Ejemplo

<https://github.com/miky-kr5/CEIDEC-Pong>



Referencias

- Scott Bilas, **A Data Driven Game Object System**,
<http://gamedevs.org/uploads/data-driven-game-object-system.pdf>
- Tom Leonard, **Thief: The Dark Project**,
http://www.gamasutra.com/view/feature/3355/postmortem_thief_the_dark_project.php
- Adam Martin, **The Future of MMOG Development**,
<http://t-machine.org/index.php/2007/09/03/entity-systems-are-the-future-of-mmog-development-part-1/>
- Varios, **The Entity-Systems Wiki**,
<http://entity-systems-wiki.t-machine.org/>
- Bad Logic Games, **Ashley Wiki**,
<https://github.com/libgdx/ashley/wiki>