

# Fundamentos de Entretenimiento Digital

*Diego C. Martínez*

Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur

# Patrones de Diseño

Los patrones de diseño **nombran, explican y evalúan**  
un **diseño importante y recurrente**  
en los sistemas de software



*Erich Gamma*



*Ralph Johnson*



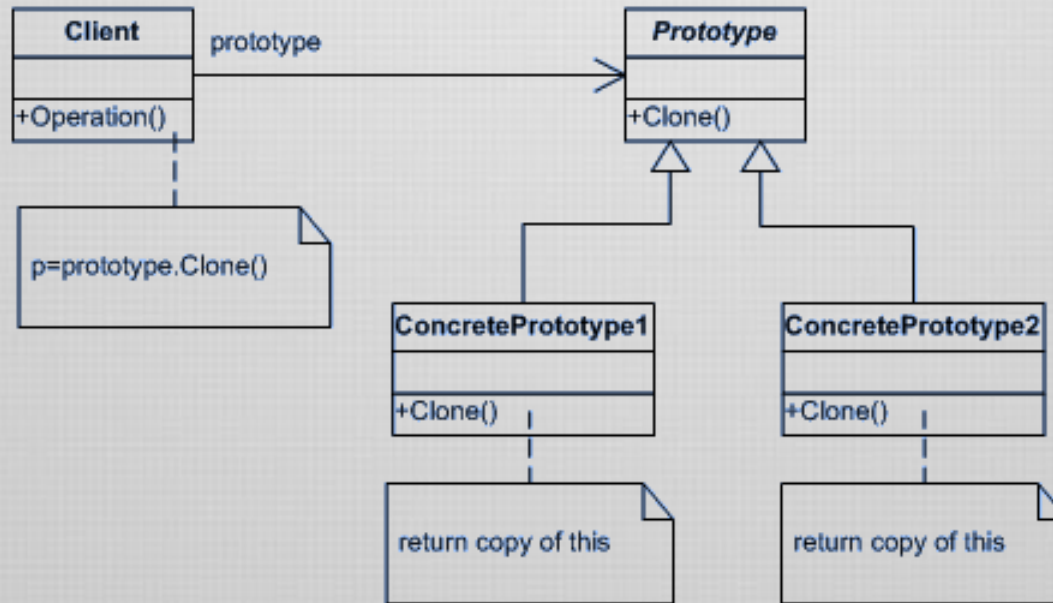
*John Vlissides*



*Richard Helm*



# Patrón Prototype



# Patrón Prototype



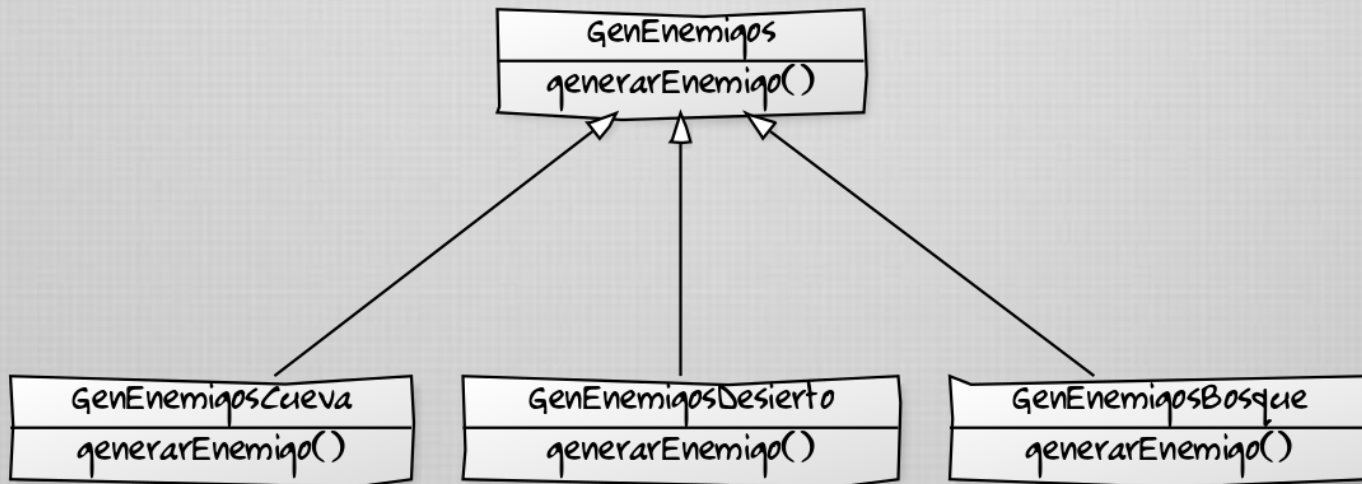


# Patrón Prototype

## Problema

Generar enemigos para cada bioma, según sean **muy** frecuentes, **algo** frecuentes o **poco** frecuentes

## Alternativa de Solución



**generarEnemigo()** genera un enemigo en una posición al azar  
Cada subclase genera mas de unos que de otros, según el bioma

# Patrón Prototype

## Problema

Generar enemigos para cada bioma, según sean **muy** frecuentes, **algo** frecuentes o **poco** frecuentes

```
GenEnemigosBosque  
generarEnemigo()
```

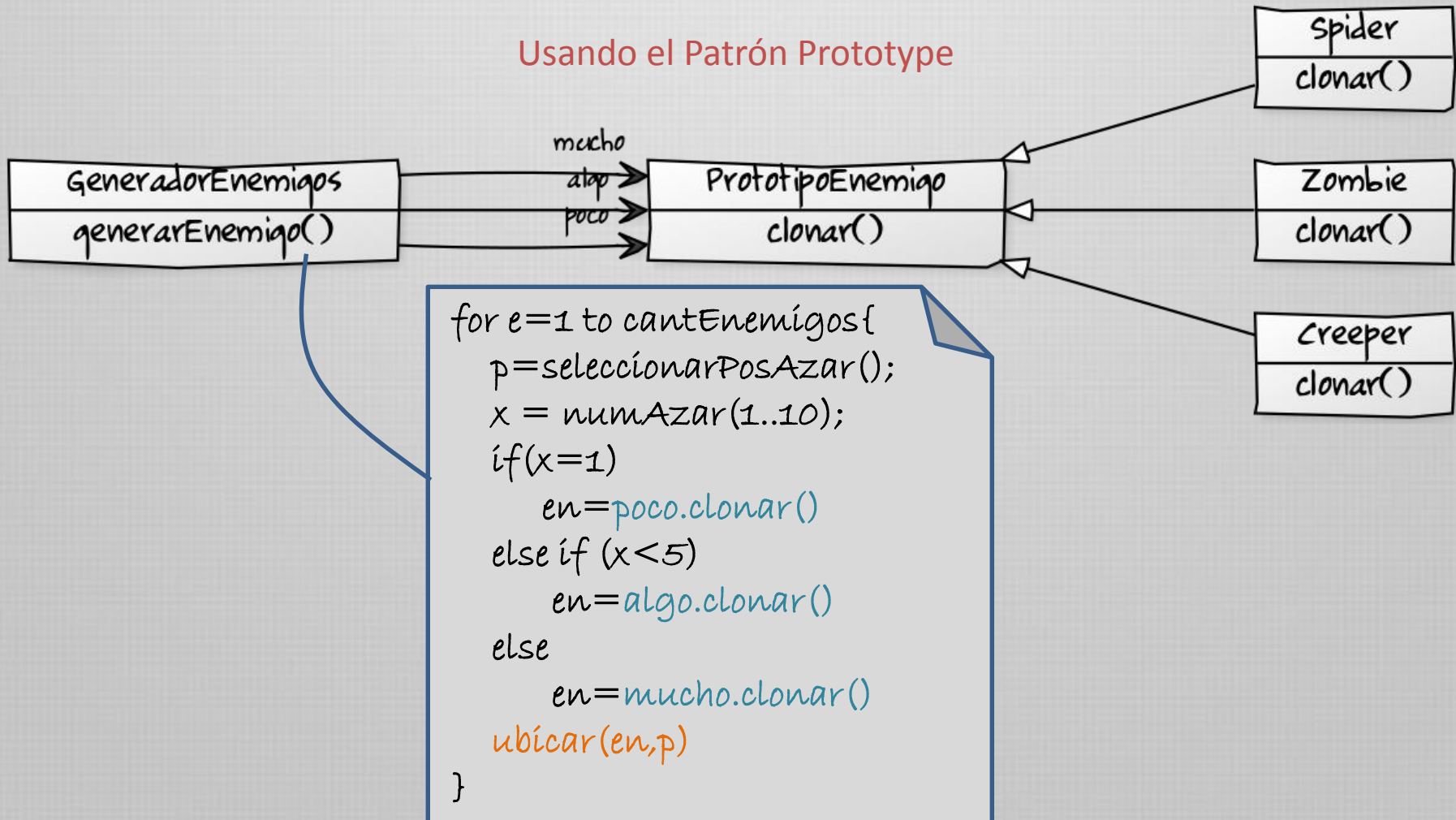
```
for e=1 to cantEnemigos{  
  p=seleccionarPosAzar();  
  x = numAzar(1..10);  
  if(x=1)  
    crearCreeper(p)  
  else if (x<5)  
    crearZombie(p)  
  else  
    crearSpider()  
}
```

# Patrón Prototype

## Problema

Generar enemigos para cada bioma, según sean **muy** frecuentes, **algo** frecuentes o **poco** frecuentes

## Usando el Patrón Prototype



# Patrón State



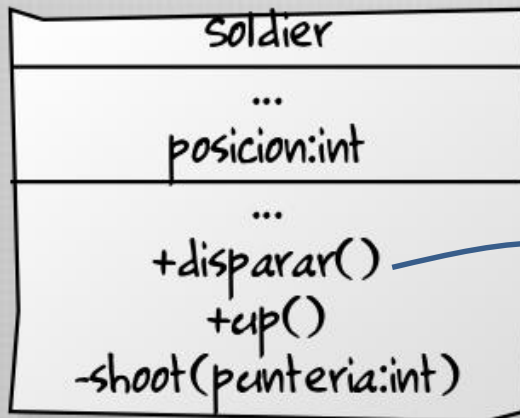
standing



kneeling



running



```
if (posición=1)//standing
then
    shoot(7);
else if (posición=2) //kneeling
then
    shoot(8)
else if (posición=3) //running
then
    shoot(2)
```



# Patrón State



standing



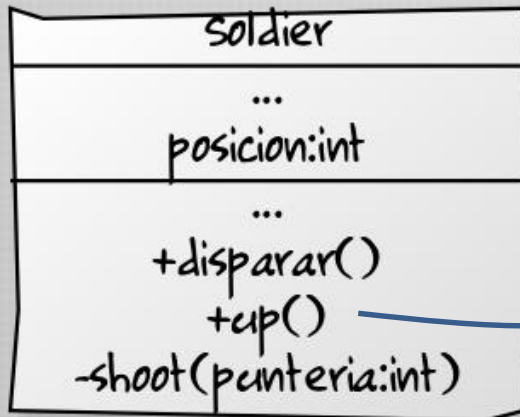
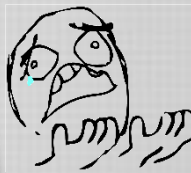
kneeling



running

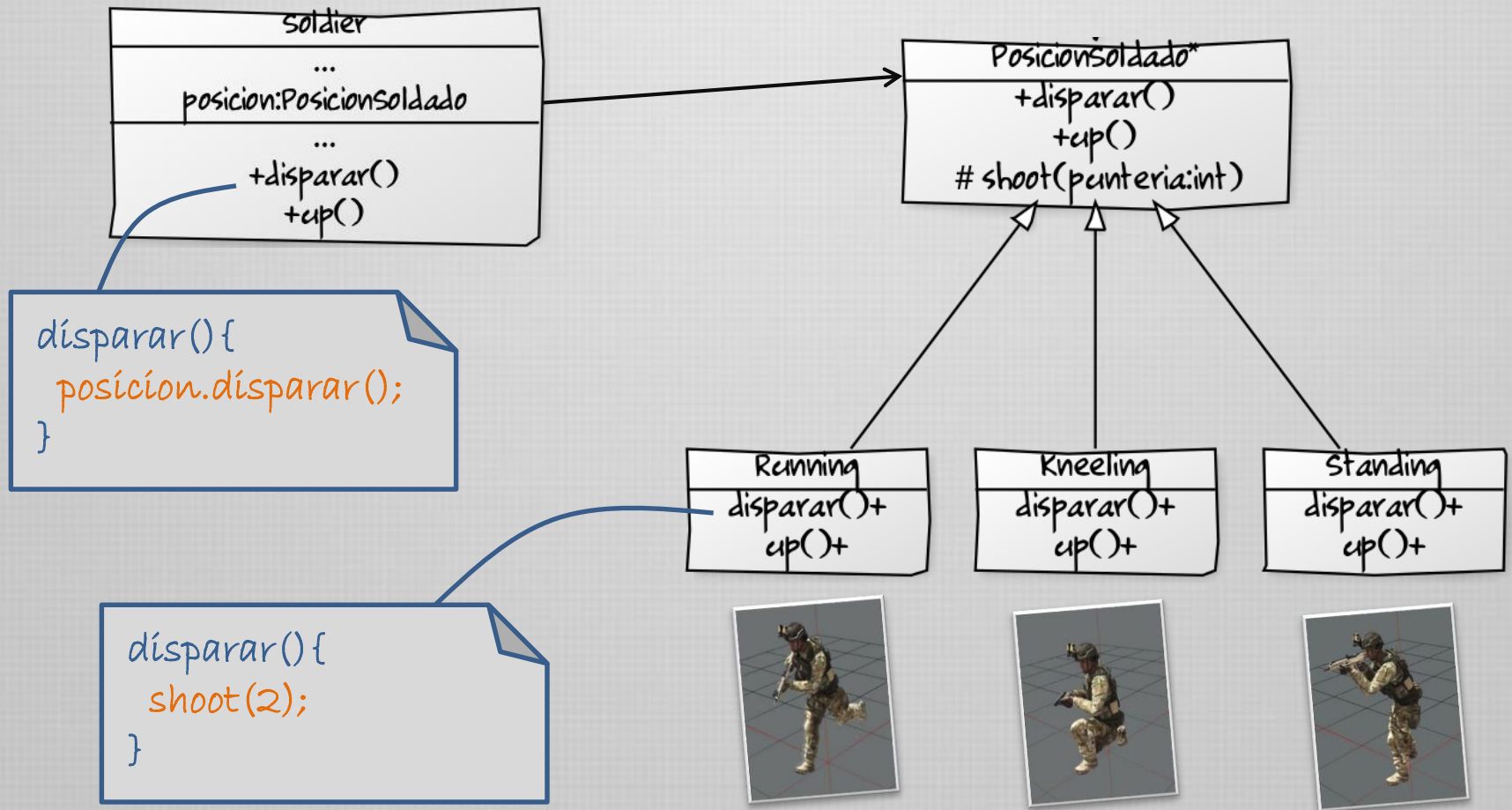


down

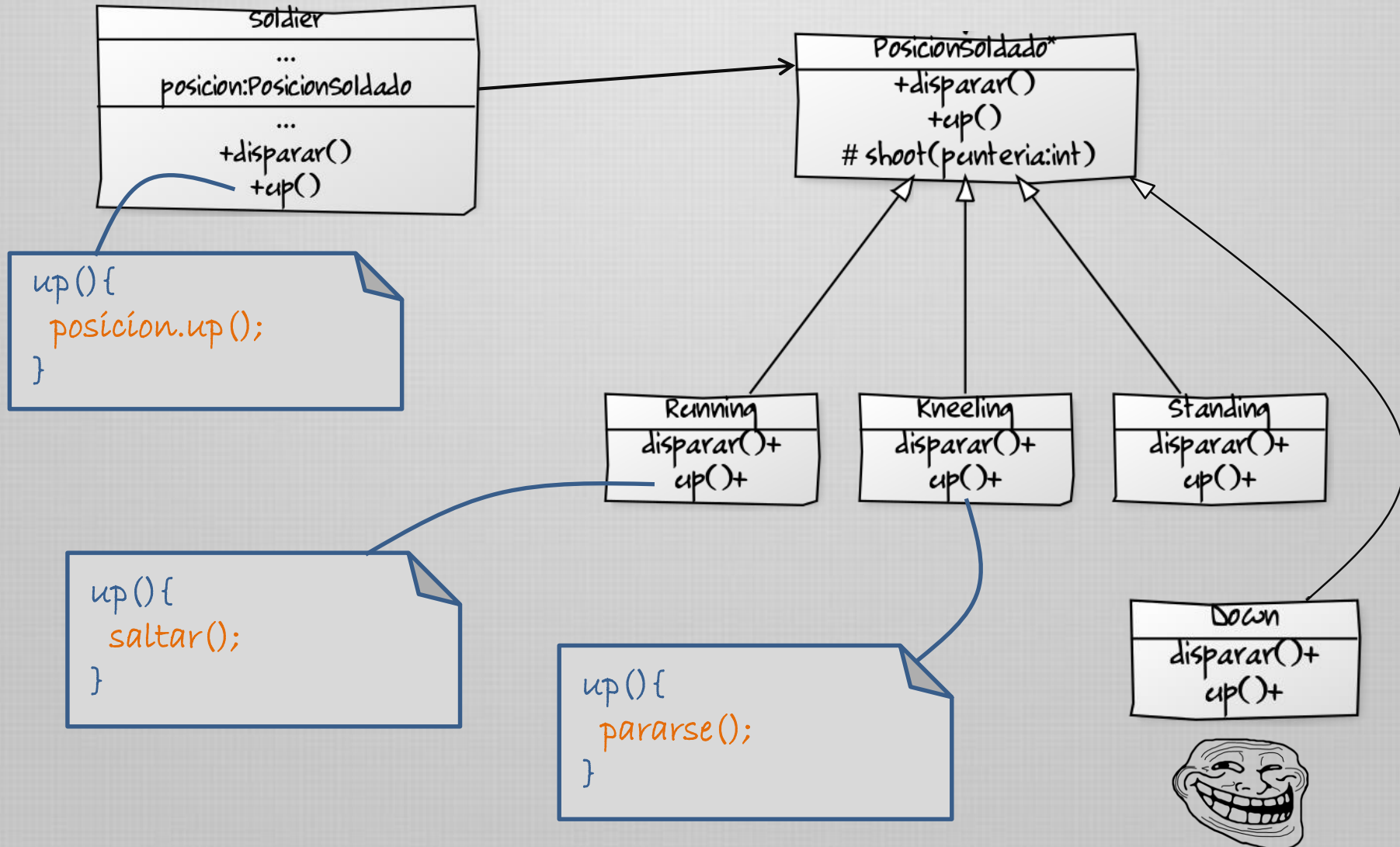


```
if (posición=1) //standing
then
  saltar()
else if (posición=2) //kneeling
then
  pararse()
else if (posición=3) //running
then
  saltar()
```

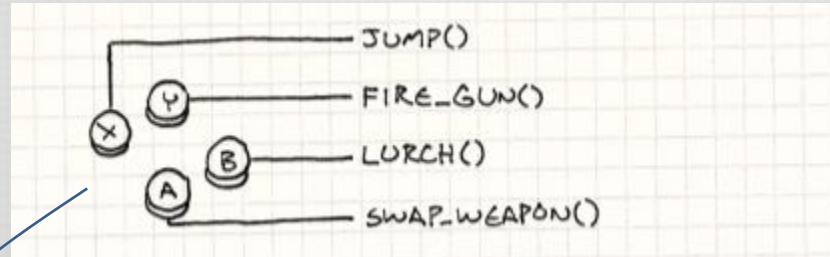
# Patrón State



# Patrón State

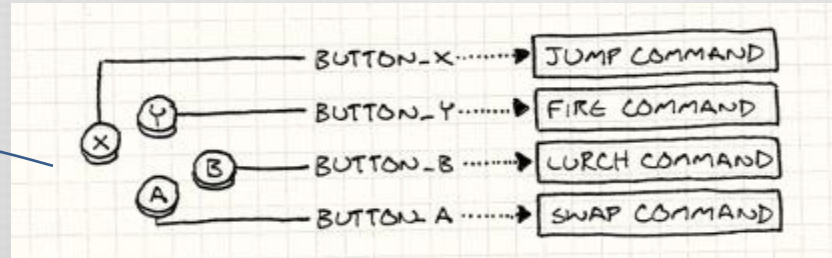


# Patrón Command



```
void InputHandler::handleInput()
{
    if (isPressed(BUTTON_X)) jump();
    else if (isPressed(BUTTON_Y)) fireGun();
    else if (isPressed(BUTTON_A)) swapWeapon();
    else if (isPressed(BUTTON_B)) lurchIneffectively();
}
```

# Patrón Command



```
class InputHandler {
```

```
...  
private:  
    Command* buttonX_  
    Command* buttonY_  
    Command* buttonA_  
    Command* buttonB_  
};
```

```
class Command {  
public:  
    virtual ~Command() {}  
    virtual void execute() = 0;  
};
```

```
class JumpCommand : public Command  
{  
public:  
    virtual void execute() { jump(); }  
};
```

```
void InputHandler::handleInput()  
{  
    if (isPressed(BUTTON_X)) buttonX_->execute();  
    else if (isPressed(BUTTON_Y)) buttonY_->execute();  
    else if (isPressed(BUTTON_A)) buttonA_->execute();  
    else if (isPressed(BUTTON_B)) buttonB_->execute();  
}
```



# Patrones

En general, todos los patrones GoF son potencialmente útiles para videojuegos



son soluciones a problemas recurrentes en el diseño de software



John Doran explica muchos de los patrones GoF en el contexto de los juegos



Robert Nystrom reconoce otros patrones no-GoF como importantes para el diseño de juegos

*Sequencing Patterns*  
*Behavioral Patterns*  
*Decoupling Patterns*  
*Optimization Patterns*



Staffan Bjork adopta otra visión de patrones diferente a la clásica de problema-solución

# Patrones

## Sequencing Patterns

Son patrones que permiten modelar aspectos del tiempo en videojuegos



Robert Nystrom



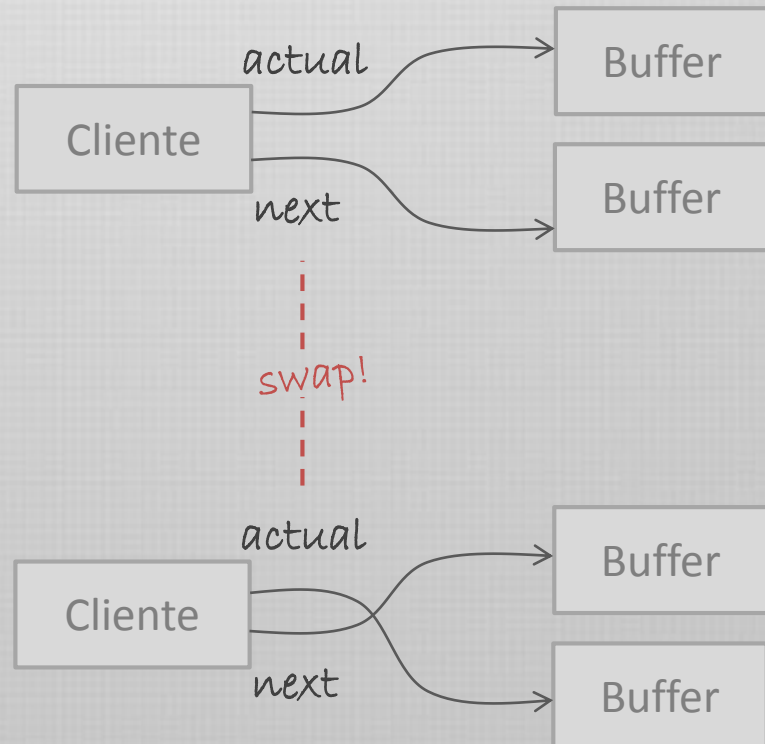
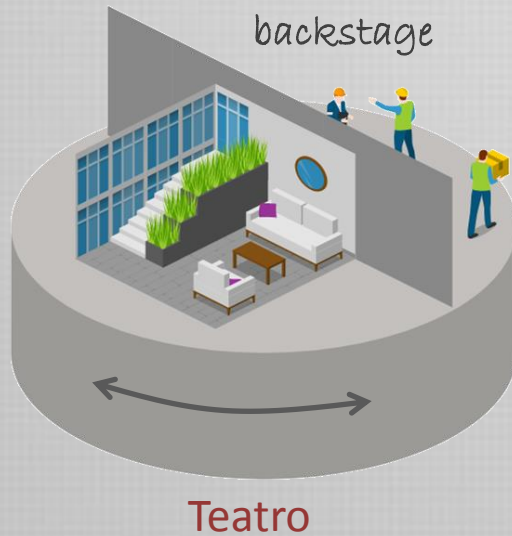
# Patrones

## Sequencing Patterns

Son patrones que permiten modelar aspectos del tiempo en videojuegos



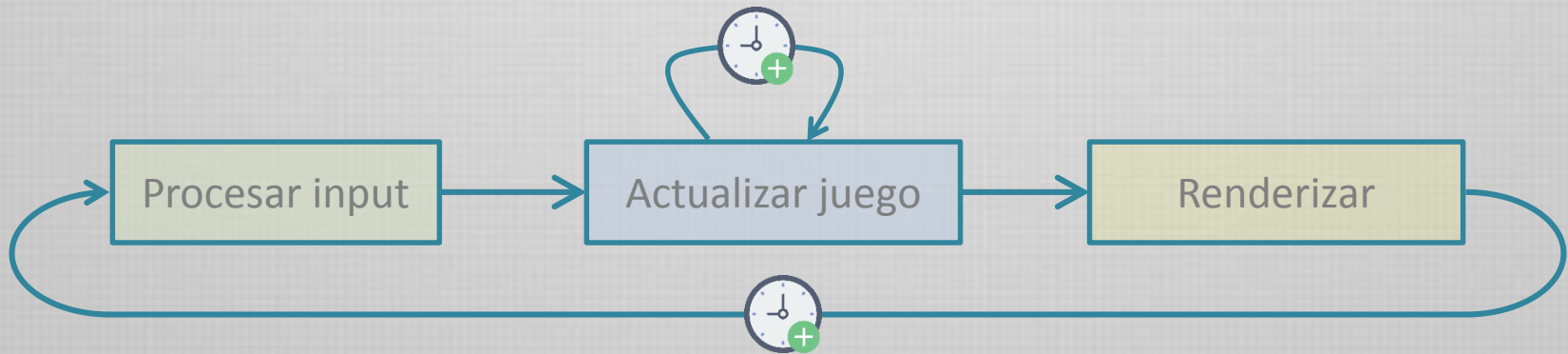
Robert Nystrom



# Patrones

## Sequencing Patterns

Son patrones que permiten modelar aspectos del tiempo en videojuegos



Robert Nystrom

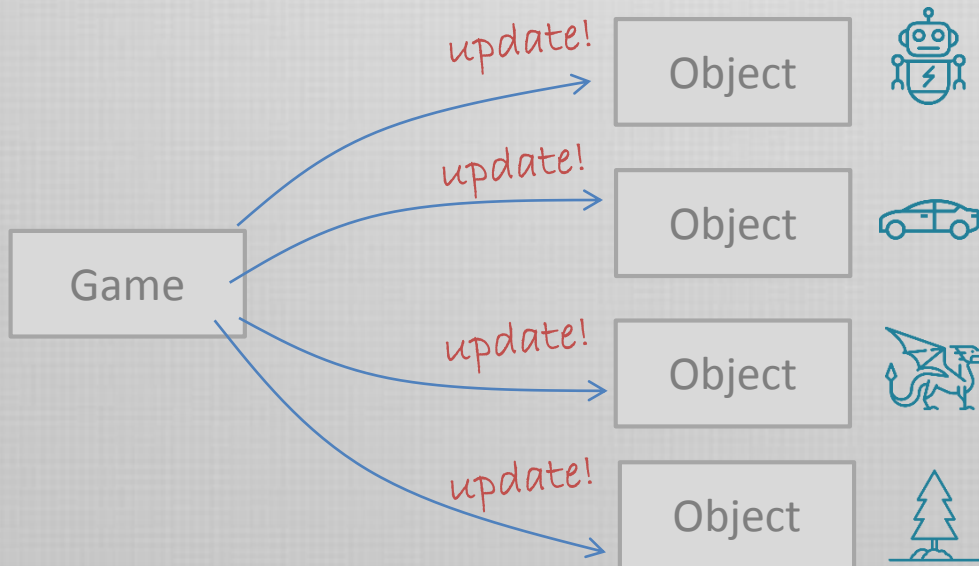
# Patrones

## Sequencing Patterns

Son patrones que permiten modelar aspectos del tiempo en videojuegos



Robert Nystrom





# Patrones

## Behavioral Patterns

Son patrones que modelan ciertos aspectos relacionados con el comportamiento de elementos del juego



Robert Nystrom



# Patrones

## Behavioral Patterns

Son patrones que modelan ciertos aspectos relacionados con el comportamiento de elementos del juego

Bytecode

Subclass Sandbox

Type Object



Robert Nystrom



```
setHealth(int wizard, in amount)
setWisdom(int wizard, in amount)
setAgility(int wizard, in amount)
playSound(int soundId)
spawnParticles(int particleType)
```



```
enum Instruction{
    INST_SET_HEALTH = 0x00,
    INST_SET_WISDOM = 0x01,
    INST_SET_AGILITY = 0x02,
    INST_PLAY_SOUND = 0x03,
    INST_SPAWN_PART = 0x04
};
```

*El efecto de un spell  
se describe ahora como  
una lista de bytes*

# Patrones

## Behavioral Patterns

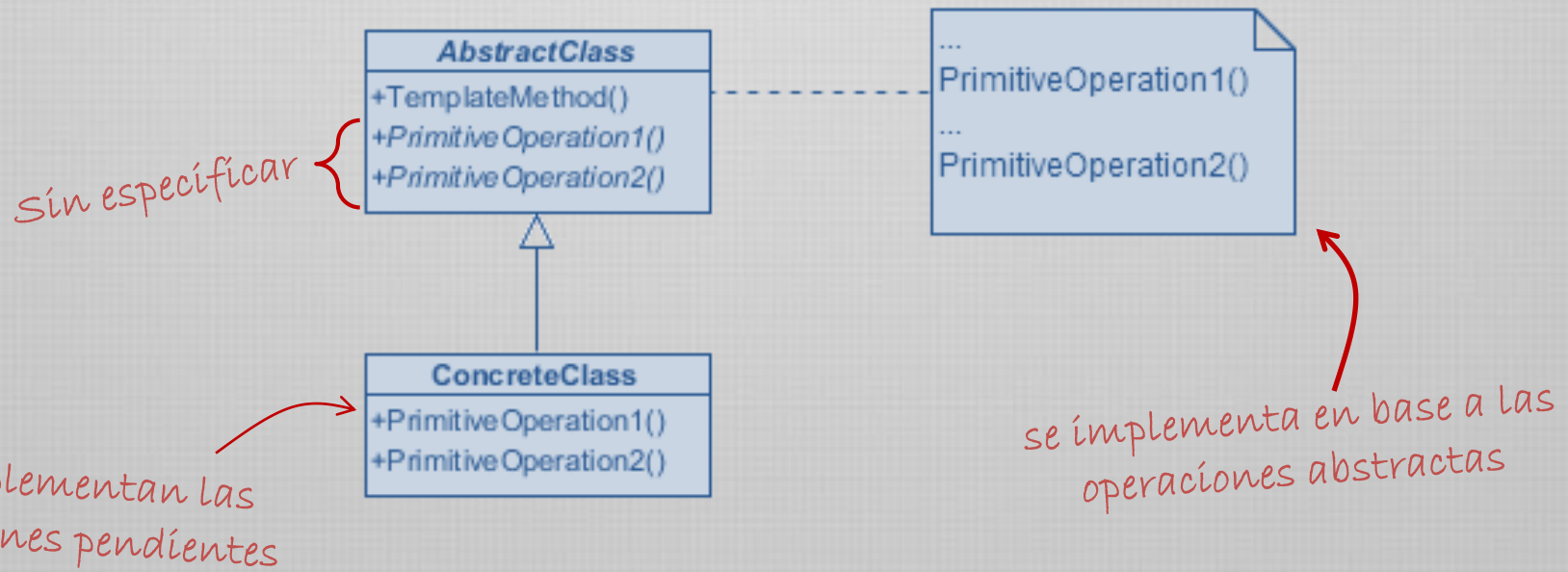
Son patrones que modelan ciertos aspectos relacionados con el comportamiento de elementos del juego

Bytecode

Subclass Sandbox

Type Object

Es similar a **Template Method** (GoF)



Robert Nystrom

# Patrones

## Behavioral Patterns

Son patrones que modelan ciertos aspectos relacionados con el comportamiento de elementos del juego

Bytecode

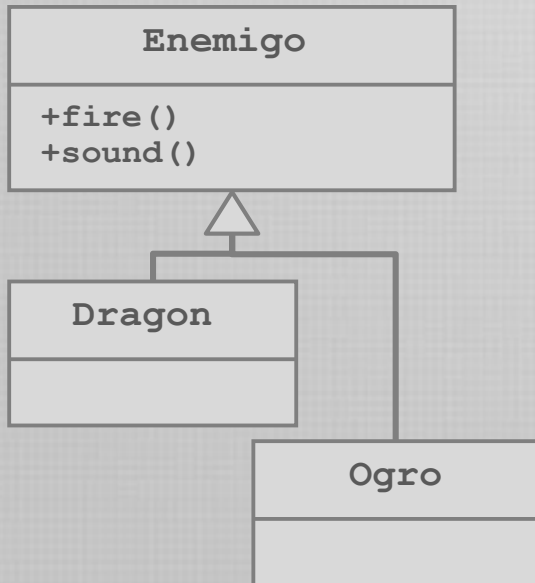
Subclass Sandbox

Type Object

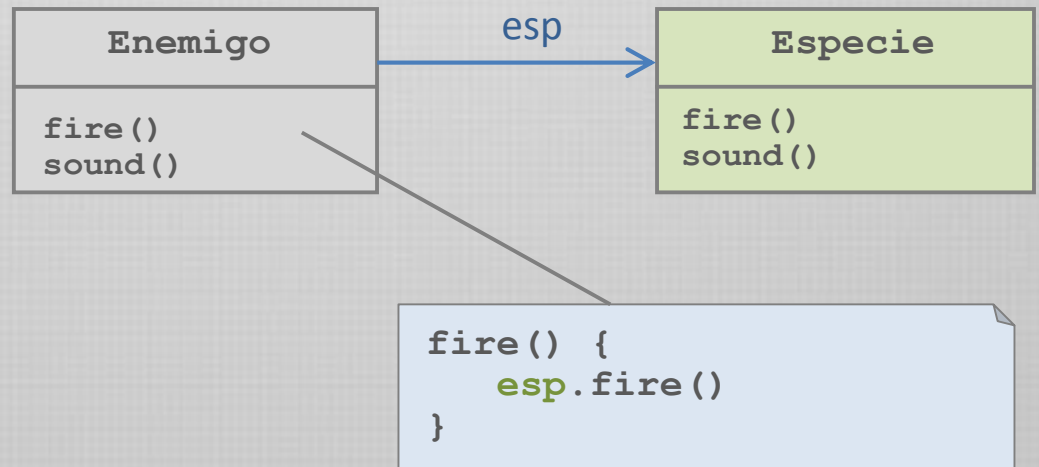


Robert Nystrom

### HERENCIA



### COMPOSICION (Type Object)



# Patrones



Robert Nystrom

Se definen también otros patrones

## Decoupling Patterns

Enfocados en  
facilitar el cambio y  
la modificación

*Component*  
*Event Queue*  
*Service Locator*

## Optimization Patterns

Enfocados en  
aumentar la  
velocidad del juego

*Data Locality*  
*Dirty Flag*  
*Object Pool*  
*Spatial Partition*

Muchos de ellos tienen fuerte similitud  
con algunos patrones GoF

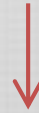


# Patrones



Staffan Bjork

En su trabajo se distinguen patrones con una visión diferente a los anteriores autores



no se basa en el esquema problema-solución



evitan tratar los patrones como soluciones a aplicar

agregar o quitar patrones afecta al gameplay  
Requiere otro enfoque

# Patrones

## Descripción de los *Game Design Patterns*

Descripción  
semiformal  
*descripción  
general del  
patrón*

Descripción  
inter-relacionada  
*relaciones del  
patrón con otros  
patrones*

Jerarquía de  
patrones  
*basada en las  
instanciaciones  
high/low-level*

Intencional o  
Emergente  
*naturaleza  
intencional  
del patrón*



*Instantiates*

*¿que otros patrones genera?*

*Modulates*

*¿que patrones afecta?*

*Instantiated by*

*¿que patrón es necesario?*

*Modulated by*

*¿qué patrones lo afectan?*

*Potential Conflicts*

*¿con qué patrones puede conflictuar?*

*Se describen con  
nombre  
definición  
descripción  
uso del patrón  
consecuencias  
relaciones  
referencias*

# Patrones

## Game Design Patterns

- for Game Elements → sobre objetos que el jugador puede manipular
- for Resource and Resource Management → sobre los tipos de recursos que existen en el juego y cómo el usuario los controla
- for Information, Communication and Presentation → sobre la información del juego y cómo se oculta o muestra al jugador
- Actions and Events → sobre acciones del jugador, como se relacionan con el estado del juego
- for Narrative Structures, Predictability, Immersion → sobre cómo el juego sustenta la inmersión
- for Social Interaction → sobre la interacción social entre los jugadores
- for Goals → sobre las metas en el juego y sus relaciones
- for Goal Structures → sobre aspectos del gameplay relacionados con metas
- for Game Sessions → sobre la participación de los jugadores en la sesión
- for Game Mastery and Balancing → sobre las habilidades del jugador necesarias, sobre el balance del juego
- for Meta Games, Replayability, Learning Curves → sobre aspectos fuera del juego

# Patrones

Los patrones se identifican en el juego por

análisis estructural

Análisis sobre el juego,  
sin necesidad de jugarlo



*Lives, High Score List, Meta Games;  
Collection, Levels, Role Reversal;  
Time Limit, Movement,  
Geometric Rewards for Investments;  
Enemies, Power-Ups, Inaccessible Areas*

play testing

Análisis sobre el juego  
al jugarlo



*Illusion of Influence  
Immersion  
Tension  
Social Interaction*

# Patrones - Game Elements

## Game Worlds

*Game World, Reconfigurable Game World, Levels, Inaccessible Areas, Consistent Reality Logic, Alternative Reality, Moveable Tiles*

## Objects

*Enemies, Boss Monsters, Deadly Traps, Obstacles, Avatars, Units, Tools, Controllers, Alarms, Pick-Ups, Power-Ups, Clues, Extra-Game Information, Invisible Walls, God's Finger, Mule, Buttons, Helpers, Traces, Resource Generators, Tiles, Dice, Cards, Card Hands, Drawing Stacks, Discard Piles*

## Abstract Objects

*Score, High Score Lists, Lives, Parallel Lives, Cameras, Ghosts, Book-Keeping Tokens*

## Locations

*Strategic Locations, Outstanding Features, Chargers, Resource Locations, Goal Points, Save Points, Spawn Points, Safe Havens*



# Patrones - Game Elements

## Game World

*El entorno en el que el juego tiene lugar está determinado por las relaciones espaciales entre los elementos del juego*

*¡Presente en casi todos los juegos!*



**Instantiates:** *Consistent Reality Logic, Immersion, Spatial Immersion, Exploration*

**Modulates:** *Roleplaying, Player Balance, Camping*

**Instantiated by:** *Tiles, Reconfigurable Game World, Levels*

**Modulated by:** *Alternative Reality, God Views, Fog of War, Game State*

*Overview, Movement, Inaccessible Areas, Tile-Laying, Game Masters, Storytelling, Never Ending Stories, Player Constructed Worlds, Dedicated Game Facilitators, Converters, Easter Eggs, Secret Resources, Obstacles, Deadly Traps, Resource Generators, Producers, Construction, Inaccessible Areas, Enemies, Converters, Resources, Goal Points, Spawn Points, Helpers, Strategic Locations, Outstanding Features, Area Control, First-Person Views, Third-Person Views, Shrinking Game World*

**Potentially Conflicting with:** *None*

# Patrones - Game Elements



## Levels

A Level is a part of the game in which all player actions take place until a certain goal has been reached or an end condition has been fulfilled.



## Boss Monsters

A more powerful enemy the players have to overcome to reach certain goals in the game.



## Pick-Ups

Pick-Ups are game elements that exist in the game world and can be collected by players, usually by moving an avatar or Units in contact with the Pick-Up.



## Inaccessible Areas

Inaccessible Areas are parts of the Game World the player can perceive but cannot currently enter, such as areas behind locked doors or sufficiently high ledges.

# Patrones - Resources



## Resources

Game elements that are used by players to enable actions in a game



## Limited Resources

The resources available to the players are limited to such extent that they are forced to plan ahead the use of the resources



## Secret Resources

Secret Resources are resources that are unknown to at least some of the players



## Symmetric Resource Distribution

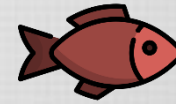
The resources are distributed symmetrically and evenly among the players, that is, the players have similar access and ownership rights to the resources

# Patrones – Information, Communication and Presentation



## Fog of War

The player has no information about game world areas that are not being observed or have not yet been explored



## Red Herrings

Information or potential goals that are designed to either mislead or distract the player



## Near Miss Indicators

Players have explicit information about how close they were to achieving a goal when they have failed to achieve it

# Patrones – Actions and Events



## Damage

Effects from actions or events that can lead to negative consequences.



## Illusionary Rewards

The player receives something that is perceived as a reward but does not quantifiably help in completing a goal in the game as expressed by the game state.



## Maneuvering

Controlling the movement of game elements in real-time games



## New Abilities

Gaining new abilities during gameplay.



# Patrones – Narrative Structures, Predictability, and Immersion



## Creative Control

Players have the ability to be creative within the Game World



## Perceived Chance to Succeed

Players believe, whether correctly or not, that they have a chance to succeed with actions in a game.



## Cut Scenes

Sequences of storytelling where players cannot act within the game.



## Identification

The characters or parts of the game with which players identify.

# Patrones – Goals



## Capture

Capture is the goal pattern where the end result is the elimination or change of ownership of an actively resisting goal object



## Delivery

Delivery consists of moving a certain game element to another specified game element or place within the game space.



## King of the Hill

Reaching and keeping a sought for game state that other players are trying to reach and keep.



## Stealth

Stealth is the goal to move through a certain area and perform an action without being detected.