

---

# Game Design Patterns and other Analytical Tools

---

staffan.bjork@chalmers.se



# But first...

- Note change in schedule, exercise today!
- Global Game Jam starts today, 15:30 here! (studios on floor 3, house Patricia)
  - *Already started* in some areas, go to the main web site to check
- Assignments #1 hopefully graded during the weekend...



---

# Structure of today's lecture

- Design Languages
- Examples of languages
  - Formal Abstract Design Tools
  - The MDA framework
  - The 400 Project
  - Game Ontology Project
  - Gameplay Design Patterns
- Using Analytical Tools



# Problems in Gameplay Design

- Explain values of novel game concepts
- Understanding differences between games
- Gain understanding within development teams
- Communication between developers and stakeholders
- Exploit new platforms and technologies
- Depersonalize intended gameplay
- Describe gameplay problems
- Specify foci of gameplay evaluations

*Notions and concepts needed - a **language** for the design of gameplay*

# Design Languages

J. Rheinfrank & S. Evenson in *Bringing Design to Software* (Ed. T. Winograd)

- Purpose and Use
  - Allows designers to *embed meaning* into artifacts
  - Allow artifacts to *express meaning* to people
    - Related to the concept of affordances
  - Allow artifacts to be *assimilated* into peoples' lives
- Components
  - Collection of elements
    - For example, the *Component Framework* from the previous lecture
  - Principles of organization
    - How the elements relate and interact with each other
  - Qualifying situations
    - When is it suitable to use components
- Gameplay design
  - Deals with an abstract and emergent feature – interaction
  - Needs to deal with both the interaction itself and that which enables the interaction

**What bad effects can rise  
from analyzing games?**

From using frameworks or design  
languages?

Examples of design  
languages?

---

# Formal Abstract Design Tools

*(articles online, e.g. gamasutra)*

---

Doug Church

*(Ultima Underworld I-II, System Shock,  
Thief I-III, Deus Ex I-II, Lara Croft Tomb  
Raider: Legend, FreQuency)*



# Formal Abstract Design Tools - Overview

- Formal
  - “implying precise definition and the ability to explain it to someone else”
- Abstract
  - “to emphasize the focus on underlying ideas, not specific genre constructs”
- Design
  - "as in, well, we're designers”
- Tools
  - "since they'll form the common vocabulary we want to create”

---

# Formal Abstract Design Tools - Examples

## *Intention*

Making an implementable plan of one's own creation in response to the current situation in the game world and one's understanding of the game play options.

## *Perceivable Consequence*

A clear reaction from the game world to the action of the player.

---

# Mechanics, Dynamics, Aesthetics

<http://algorithmancy.8kindsoffun.com/>

---

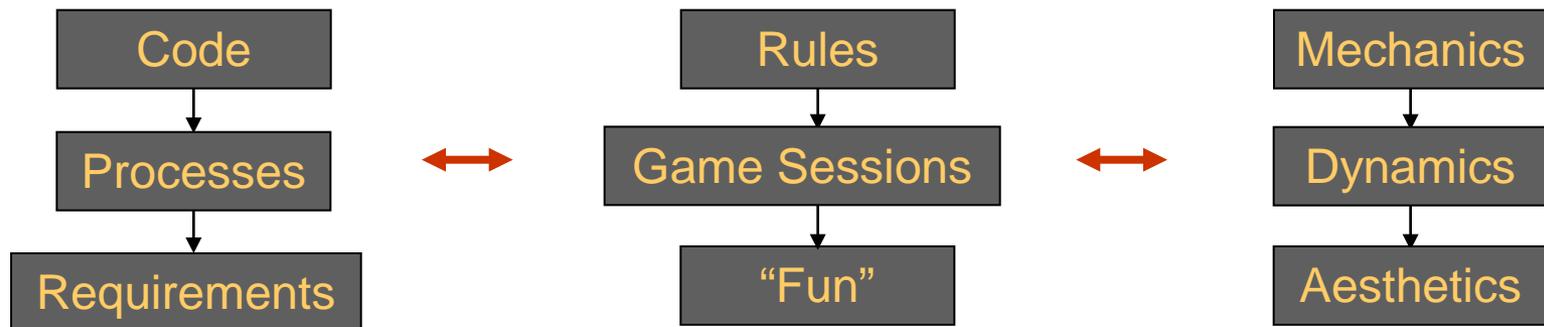
Marc LeBlanc

*(Ultima Underworld II, System Shock, Flight Unlimited, Terra Nova, Thief I-II, Deus Ex, NFL 2K2, NBA 2K2, Oasis, Field Commander)*



# MDA - overview

- Games are state machines
  - Games are programs



# MDA – Comments about aesthetics

- “We need to understand the *emotional* requirements of our software”
  - Fun, challenge, sense of achievement, sorrow, frustration
- Regarding requirements
  - “With productivity software, *the user brings his goals to the application*”
  - “With games, *the application brings goals to the user*”
- Regarding goals
  - “As designers, we can choose certain *aesthetics as goals* for our game design”
    - Aesthetics of gameplay?
  - “As with other software, our process is driven by *requirements*, not *features*”

# MDA - Eight Kinds of "Fun"

## 1. Sensation

*Game as sense-pleasure*

## 2. Fantasy

*Game as make-believe*

## 3. Narrative

*Game as drama*

## 4. Challenge

*Game as obstacle course*

## 5. Fellowship

*Game as social framework*

## 6. Discovery

*Game as uncharted territory*

## 7. Expression

*Game as self-discovery*

## 8. Submission

*Game as pastime*

**How does the MDA model  
support analyzing games?  
Designing games?**

---

# 400 project

[http://www.theinspiracy.com/400\\_project.htm](http://www.theinspiracy.com/400_project.htm)

---

Noah Falstein

*(Maniac Mansion, Secret Weapons of the Luftwaffe, The Secret of Monkey Island, Loom, Indiana Jones and The Last Crusade: The Graphic Adventure, Monkey Island 2: LeChuck's Revenge, Indiana Jones and the Fate of Atlantis, Star Wars: Empire at War, ParaWorld)*



# 400 Project - Overview

- Help Game Designers by providing them with rules
  - *Normative*
  - Best Practice description
- Examples
  - Fight Player Fatigue
  - Make Subgames
  - Begin at the Middle
  - Make Challenges Vary in More than Degree
  - Provide Both Safe and Dangerous Areas
- 400?
  - “That’s just a rough number, ...”

# 400 Project - Format

- [Name]
- A concise, imperative statement of the rule, both as a sentence and paragraph
- Its domain of application
  - (both its hierarchy, e.g. a rule about rules, a rule about the development process, or just a rule about games themselves, and genre, e.g. Applies only to RTS games or Online games).
- Rules or circumstances that it trumps
  - over which this rule takes precedence)
- Rules or circumstances that it is trumped by
- An example or two from well-known published games, if applicable, as well as counter-examples that show the consequences of not following the rule

# 400 Project - Example

## *Provide Clear Short-Term Goals*

### Description

Always make it clear to the player what their short-term objectives are. This can be done explicitly by telling them directly, or implicitly by leading them towards those goals through environmental cues. This avoids the frustration of uncertainty and gives players confidence that they are making forward progress.

### Domain

This is a basic rule of game design, and applies to all games directly.

### Trumps

It trumps the rule “Emphasize Exploration and Discovery” because the player should not have to discover their short-term goals. If discovery is warranted, it should be to discover the tools or information needed to achieve the clear, short-term goals, not to discover the goals themselves. It also trumps “Provide an Enticing Long-Term Goal”, as it is more important to have the player know what to do next than to simply know that they have to Kill the Evil Wizard/Save the World/Rescue the Princess.

### Trumped by

It is trumped by the rule “Make the First Player Action in a Game Painfully Obvious”. However, often that first obvious action in a game – read the paper, click on the wise old man, shoot the monster – should trigger an explanation of the first short-term goal beyond that.

### Examples

When Hal Barwood and I designed Indiana Jones and the Fate of Atlantis we gave the player explicit goals throughout the game by having the supporting characters guide the objectives. The initial theft of an artifact by a Nazi agent led the player (in the role of Indiana Jones) to Madam Sophia, who in turn presented Indy with his next objective, and so on. One short-term goal, like “convince this character to give you an artifact”, often triggered conversation with the character that led to the next goal, like “find the lost dialog of Plato”.

Shigeru Miyamoto uses clear short-term goals throughout all of his games. In Mario 64 he uses explicit goals like characters or signs that tell you how to move, jump or swim, adjacent to appropriate obstacles. Other goals are implicit ones, as when you’re left to explore the landscape at the beginning of the game with a large castle dominating the landscape and a drawbridge leading right to it. He also uses strings of floating coins to pick up as implicit goals that help lead the player into attempting jumps and using catapults or cannons pointing toward the coins.

More recently, Halo from Bungie does an admirable job of using the landscape itself and suggestions from both an AI companion and fellow Marines to channel you towards the next short-term goal.



# 400 Project - Current Status

- Work in progress
  - 112 rules in list
  - 2 described accord to format
    - Others in 250 words or less
  - Contributors from several professionals
    - Sid Meier, Raph Koster, Warren Spector,
    - Albert Einstein...

[http://www.theinspiracy.com/400\\_project.htm](http://www.theinspiracy.com/400_project.htm)



**Is it good or bad to have  
rules on how you should  
design?**

Does it support analyzing games?

---

# Game Ontology Project

*<http://www.gameontology.org>*

---

Mateas M., Zagal, J. &  
Fernandez, C.



# Game Ontology Project - Overview

- Ontology
  - “a branch of metaphysics concerned with the nature and relations of being” and “a particular *theory about* the nature of being or the kinds of *things that have existence*”
  - Identifies important structural elements
  - Relationships between elements
  - Organizes these hierarchically
    - Parent- Child relation
- Top Levels in the hierarchy
  - Interface
  - Rules
  - Entity Manipulation
  - Goals

# Game Ontology Project - Format

- Category: Name
- Examples
  - Strong example
  - Weak example
- Relations
  - Parent
  - Children
- References

# Game Ontology Project - Example

## *Locus of Manipulation*

A games locus of manipulation is where the players ability to control and influence the game is located. In many games, the players manipulative powers are tied to either an on-screen or implied avatar, such as the on screen representation of Mario in Super Mario Sunshine (Koizumi and Usui, 2002) or an implied player avatar like in Doom (Carmack, 1993). In other games it is tied to a number of entities, whether anthropomorphic, as in Warcraft III (Pardo, 2002) or more object like, such as the tetrads in Tetris (Pajitnov, 1986). In all of these cases, at any given moment of play, the player exerts control over some game entity or entities, but not over others.

Secondarily, the locus of manipulation provided within a game can work with other aspects of the games presentation and rules to create a sense of identification between the player and the role he plays within a game, or Player Position (Costikyan, 1994). This is especially true in games where the player controls an avatar or a group of anthropomorphic entities. In Super Mario Sunshine (Koizumi and Usui, 2002), the game centers the players control and view of the world on Mario so as to lead the player to identify with Mario. In Madden NFL 2004 (Tiburón, 2003), the player is led to identify with the team he is playing, either as a team, favorite players, or in the capacity of coach. The game provides presentational and subgame modes to reinforce each position.

## Parent

- \* Input Method

## Children

- \* Multiple Entity Manipulation
- \* Single Entity Manipulation

## References

- Carmack, J. (1993). Doom. id Software, dos edition.
- Costikyan, G. (1994). I have no words and I must design. Interactive Fantasy, (2).
- Koizumi, Y. and Usui, K. (2002). Super Mario Sunshine. Nintendo, gamecube edition.
- Pajitnov, A. (1986). Tetris. Dos edition.
- Pardo, R. (2002). Warcraft III: Reign of Chaos. Blizzard Entertainment, windows edition.
- Tiburón, developer (2003). Madden NFL 2004. Electronic Arts, xbox edition.

# Game Ontology Project – Current Status

- About ~200 entries
- Wiki-based project
  - Involve the gamer community
  - That is developed by players
- Describes games from the player's perspective
- Does not seem to have been update for quite some while

<http://www.gameontology.org>

**Do players provide a good  
or bad basis for developing  
an ontology?**

How does it support analyzing or  
designing games?

---

# Gameplay Design Patterns

*[www.gamedesignpatterns.org](http://www.gamedesignpatterns.org)*

*[www.gameplaydesignpatterns.org](http://www.gameplaydesignpatterns.org)*

---

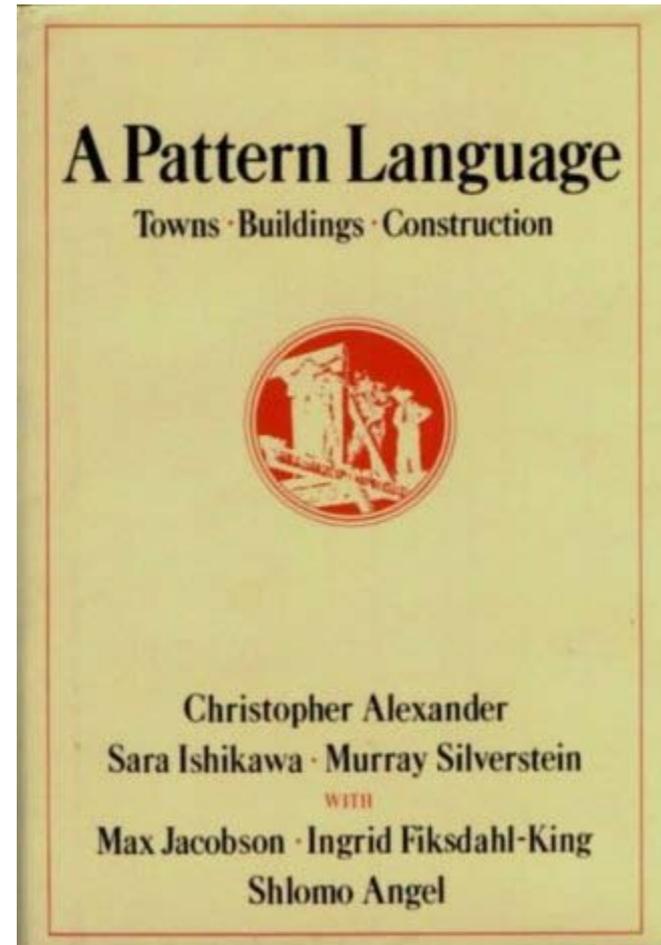
Staffan Björk & Jussi Holopainen

*(Not any games you would know about)*



# Origin of Design Patterns

- Patterns of design within architecture
  - “The Quality Without a Name”
  - Re-Use allow accumulation and generalization of solutions
  - Allow all members of a community or design group to participate
  - Framed as pairs of problems and solutions
  - Embedded ideology



# One View on Design Patterns

- A way to describe reoccurring design choices
  - Offers possible explanations to why these design choices have been made
  - Codify unintentional features so they can be intentional choices in later designs
- A guide of how to make similar design choices in game projects
  - What is required to make a pattern emerge
  - What consequences do a pattern have?
  - Not only problem solving
- Game Design Patterns a way to describe components on all levels within the design language

## PATTERNS *in* GAME DESIGN

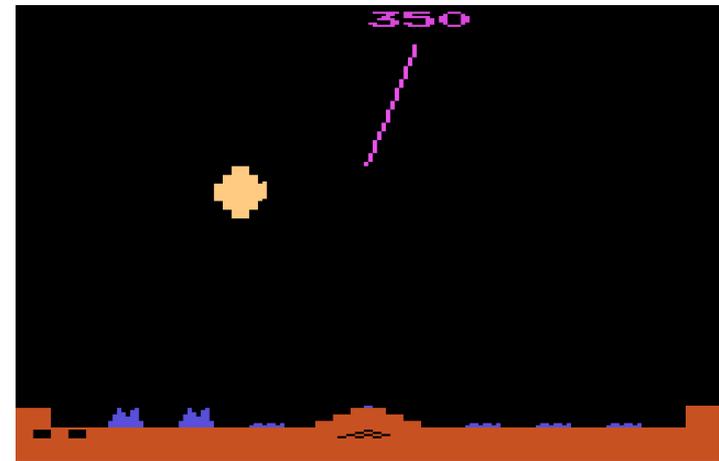
- Provides professional and aspiring game designers with the tools they need to focus on game play while designing games
- Teaches the basic concepts of game play through the study of design patterns
- Includes patterns that are usable as they are or customizable as needed
- Emphasizes design methodology in a style that is independent of the technology chosen
- Includes a CD-ROM with an easy-to-use collection of patterns (200+) on multiple platforms, prototype games and their accompanying design documents, background material for a national workshop, and samples



STAFFAN BJÖRK / JUSSI HOLOPAINEN

# Game Design Pattern Examples

- Power-Ups
- Boss Monster
- Paper-Rock-Scissor
- Cut Scenes
  
- Role Reversal
- Parallel Lives
- Orthogonal Unit Differentiation
- Social Interaction



# Game Design Pattern - Format

- Name
- Introduction
  - One line description
  - Short stand-alone description
- Examples
- Using the Pattern
- Consequences
- Relations
  - Instantiates – Instantiated by
  - Modulates – Modulated by
  - Possibly Conflicting with

# Producer- Consumer, cont.

## Description

*The production of resource by one game element that is consumed by another game element or game event*

Producer- Consumer determines the lifetime of game elements, usually resources, and thus governs the flow of the game play.

Games usually have several overlapping and interconnected Producer- Consumers governing the flow of available game elements, especially resources. As resources are used to determine the possible player actions these Producer- Consumer networks also determine the actual flow of the game play. Producer- Consumers can operate recursively, i.e. one Producer- Consumer might determine the life time of another Producer- Consumer. Producer- Consumers are often chained together to form more complex networks of resource flows.

# Producer- Consumer, cont.

Example:

In Civilization the units are produced in cities and consumed in battles against enemy units and cities. This kind of a Producer- Consumer is also used in almost all real-time strategy games.



Example:

In Asteroids the rocks are produced at the start of each level and are consumed by the player shooting at them. The same principle applies to many other games where the level progression is based on eliminating, i.e. consuming, other game elements: the pills in Pac-Man, free space in Qix, and the aliens in Space Invaders.



# Producer- Consumer, cont.

## Using the pattern

As the name implies, *Producer-Consumer* is a compound pattern of *Producer* and *Consumer* and as such this pattern governs how both of these are instantiated. The effect of producing and consuming *Resources* or *Units* often turns out to be several different pairs of *Producer-Consumers* as the produced game element can be consumed in many different ways. For example, the *Units* in real-time strategy game such as the *Age of Empires* series can be eliminated in direct combat with enemy *Units*, when bombarded by indirect fire, and finally when their supply points are exhausted. The *Producer-Consumer* in this case consists of the *Producer* of the *Units* with three different *Consumers*.

# Producer- Consumer, cont.

Using the pattern (cont.)

*Producer-Consumers* are often, especially in *Resource Management* games, chained together with *Converters* and sometimes *Containers*. These chains can in turn be used to create more complex networks. The *Converter* is used as the *Consumer* in the first *Producer-Consumer* and as the *Producer* in the second. In other words, the *Converter* takes the resources produced by the first *Producer* and converts them to the resources produced by the second *Producer*.

This kind of *Producer-Consumer* chains sometimes have a *Container* attached to the *Converter* to stockpile produced *Resources*. For example, in real-time strategy game *StarCraft* something is produced and taken to the converter and then converted to something else and stockpiled somewhere. Investments can be seen as *Converters* that are used to convert *Resources* into other forms of *Resources*, possibly abstract ones.

# Producer- Consumer, cont.

## Consequences

As is the case with the main subpatterns *Producer* and *Consumer* of *Producer-Consumer*, the pattern is quite abstract but the effects on the flow of the game are very concrete. The *Producer-Consumers* simply govern the whole flow of the game from games with a single *Producer-Consumer* to games with complex and many layered networks of *Producer-Consumers*.

# Producer- Consumer, cont.

## Consequences

The feeling of player control is increased if players are able to manipulate either the *Producer* or the *Consumer* part or both. However, in more complex *Producer-Consumer* chains this can lead to situations where players lose Illusions of Influence as the effects of individual actions can become almost impossible to track down and the process no longer has *Predictable Consequences*. Also, adding new *Producer-Consumers* that the players have control over gives them opportunities for more *Varied Gameplay*. *Producer-Consumer* networks with *Converters* and *Containers* are used in *Resource Management* games to accomplish the *Right Level of Complexity*. The game usually starts with simple *Producer-Consumers* and as the game progresses new *Producer-Consumers* are added to the network to increase the complexity.

# Producer- Consumer, cont.

## Relations

Instantiates: Varied Gameplay, Resource Management

Modulates: Resources, Right Level of Complexity, Right Level of Difficulty, Investments, Units

Instantiated by: Producers, Consumers, Converters

Modulated by: Container

Potentially Conflicting with: Illusions of Influence, *Predictable Consequences*

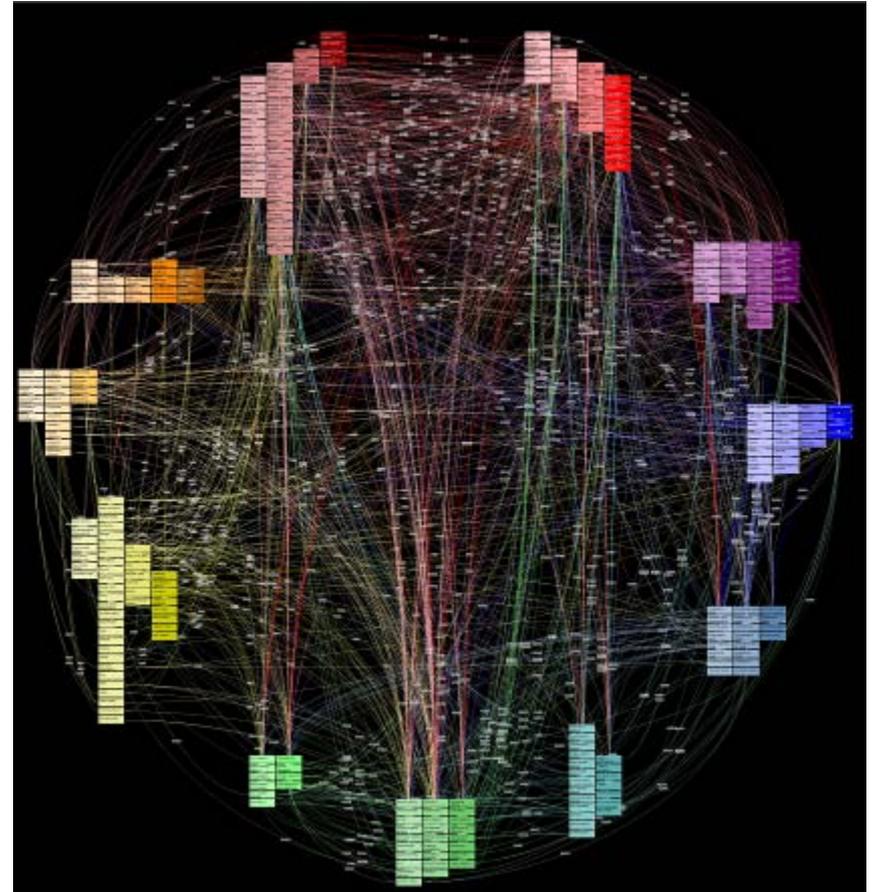


# Advantages of Design Patterns

- Allow definitions of “fuzzy” concepts
- Allow network of relations between the concepts
- Allow perspectives for both analysis and design
- Allow different levels of abstraction
- Do not require specific methods
- Specific or own collection of design patterns can be created
- Describe games from a systems (or structural) perspective

# Disadvantages of Design Patterns

- “Fuzzy” concepts
- Large collection
- Learning curve
- Usability threshold
- Developed only for gameplay design
  - Not all design disciplines needed to make a game
  - Does not describe games from the players’ perspective
    - Is this bad?



# Design Patterns – Current Status

- Large collection
  - ~ 300 patterns described and cross-referenced
  - ~ 50 new patterns to be incorporated
    - Patterns for objects in MMOGs
    - Patterns for gameplay features in MMOGs
    - Patterns for Pervasive Games
    - Patterns for Character design
    - Patterns for Dialogue Systems in Games
- Both an approach to gameplay design and a specific collection

# Exercise: What design patterns exist in Tetris?

*Not a quiz on the patterns identified by Björk & Holopainen!*

**How does design patterns  
support analyzing games?  
Designing games?**

---

# Using Analytical Tools

---



# Using Analytical Tools

- Supports methodical work
  - Support having complete overview
  - Allows finding anomalies
  - Ease use of being objective
- Supports shared understanding
  - Helps readers understand
    - Common vocabulary
- About using Tools
  - Do not solve problem by simply applying them
    - Support first (mechanical) comparison
  - Requires a focus by the tool users
    - Goal or hypothesis

---

# Accessibility of the Tools

- Most available online
  - Links from course homepage
- For patterns
  - Ask Staffan
  - But you might as well create your own mini collection highlighting 2- 3 main patterns
    - *Especially for the assignments in this course!*

# Regarding Assignment 2

- Mandatory to *identify* gameplay design patterns ideas
- But find your *own suggestions* for patterns
- Suggestion: use the component framework to make a comprehensive search of the games
  - But do not include all these patterns thus found since only a few are relevant to your question



---

# Thank you!

---

Questions?

