Data-Driven and Component-Based Game Entities

Thomas Kinnen

February 5, 2012
Overview

Introduction
  About Me
  Motivation

Component-Based
  Other Architectures
  Components Explained

Data-Driven
  Definition

Real Examples
  0 A.D.
  Unknown Horizons

Summary
About Me

- Thomas "nihathrael" Kinnen
- Pursuing Masters in computer science at TU München, Germany
- Project coordinator and programmer at Unknown Horizons
Unknown Horizons

www.unknown-horizons.org

Thomas Kinnen
Data-Driven and Component-Based Game Entities
<table>
<thead>
<tr>
<th>Introduction</th>
<th>Component-Based</th>
<th>Data-Driven</th>
<th>Real Examples</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Motivation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Architecture in games? You gotta be kiddin’ me!**

Motivation

Three good reasons for a nice architecture:

▶ Easier to maintain/extend
▶ Allows separation of labor
Motivation

Three good reasons for a nice architecture:

▶ Easier to maintain/extend
▶ Allows separation of labor
▶ Easy to add/change content
Motivation

Three good reasons for a nice architecture:

- Easier to maintain/extend
- Allows separation of labor
- Easy to add/change content
Component-Based

What does component-based mean and why do we need it?
Example of multiple-inheritance based entities
Disadvantages

- Difficult to test
- Therefore difficult to maintain and extend
Disadvantages

- Difficult to test
- Therefore difficult to maintain and extend
- Problems with member variable naming
Disadvantages

- Difficult to test
- Therefore difficult to maintain and extend
- Problems with member variable naming
- Problems with method naming
Disadvantages

- Difficult to test
- Therefore difficult to maintain and extend
- Problems with member variable naming
- Problems with method naming
- Very high complexity of tracebacks
Disadvantages

- Difficult to test
- Therefore difficult to maintain and extend
- Problems with member variable naming
- Problems with method naming
- Very high complexity of tracebacks
Definition

A **component** is a class handling a single functionality with as few dependencies on other code/components as possible.
Idea

We favor composition over inheritance.

→ Functionality by adding components using composition, not inheritance.
Details

Entity = unique ID + a set of components.

Example Spearman:
1337: (Attack, VisualActor, Cost, Position)

Figure: 0 A.D. spearman artwork
## Advantages

- Easy to test
- Very flexible

Advantages

- Easy to test
- Very flexible
- Small separate blocks → division of labor

Image from: http://www.officialpsds.com/Two-Thumbs-Up-Frog-PSD40156.html
Advantages

- Easy to test
- Very flexible
- Small separate blocks → division of labor
  - Works perfectly together with the data-driven approach

Image from: http://www.officialpsds.com/Two-Thumbs-Up-Frog-PSD40156.html
Advantages

- Easy to test
- Very flexible
- Small separate blocks → division of labor
- Works perfectly together with the data-driven approach

Image from: http://www.officialpsds.com/Two-Thumbs-Up-Frog-PSD40156.html
Component Communication

Message Bus receives messages and sends them to registered receivers.

Component Communication Diagram:
- **Component** sends message to **Message Bus**.
- **Message Bus** forwards the message.
- **Statistics Module** receives the message.
- **GUI** also receives the message.
- **Receiving Component** is connected to the message bus and awaits messages.
Data-Driven

What does data-driven mean?
Definition 1/3

data affects code’s behaviour
Definition 2/3

Separate data from code

Example:
Save unit’s health into a database/file, not in the code.
Definition 2/3

Separate data from code

Example:
Save unit’s health into a database/file, not in the code.
Definition 3/3

Game entities are defined by data, not code
0 A.D.

http://wildfiregames.com/0ad/

Thomas Kinnen
Data-Driven and Component-Based Game Entities
High-Level Architecture
A unit definition

```xml
<Entity parent="units/cart_cavalry_spearman_b">
  <Attack>
    <Hack>6.0</Hack>
  </Attack>
  <Health>
    <Max>140</Max>
  </Health>
  <UnitMotion>
    <WalkSpeed>9.68</WalkSpeed>
  </UnitMotion>
  <VisualActor>
    <Actor>units/cavalry_spearman_a.xml</Actor>
  </VisualActor>
</Entity>
```
Unknown Horizons

www.unknown-horizons.org

Thomas Kinnen
Data-Driven and Component-Based Game Entities
Making the Transition

Moving a big, **existing and working** system to a new architecture requires:

- Very good knowledge of the code
- Time
- More Time

Strategy

Unknown Horizons

Data-Driven and Component-Based Game Entities
A unit definition

```plaintext
baseclass: fightingship.FightingShip
id: 1000020
name: Frigate
radius: 5
velocity: 12.0
components:
- StorageComponent:
  inventory:
    PositiveTotalNumSlotStorage:
      limit: 120
      slotnum: 4
- ShipNameComponent
- HealthComponent:
  maxhealth: 200

actionsets:
  as_frigate0: {level: 0}
```

Unknown Horizons
Thomas Kinnen
Data-Driven and Component-Based Game Entities
Summary

- Data-driven architectures ease content creation
- Component-driven architectures are flexible and easy to test and maintain/extend
Summary

- Data-driven architectures ease content creation
- Component-driven architectures are flexible and easy to test and maintain/extend
- The combination of both separates game-design and code as much as possible
Summary

- Data-driven architectures ease content creation
- Component-driven architectures are flexible and easy to test and maintain/extend
- The combination of both separates game-design and code as much as possible
Questions

Questions and Feedback
(and advertisement)

www.unknown-horizons.org