

DESIGN PATTERNS: STRUCTURAL

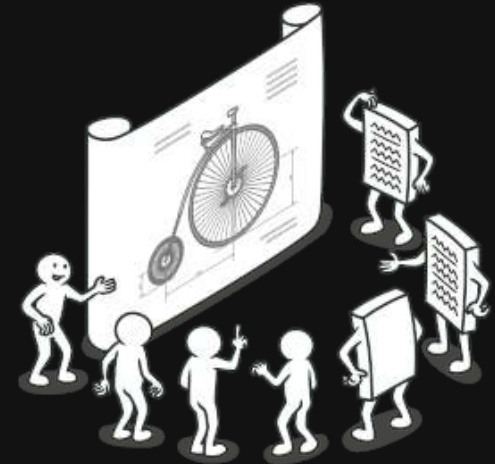


PROGRAMMA

1. Design Patterns; Wat?
2. Structural Design Patterns; Wat?
3. SDP; Proxy Pattern
4. SDP; Flyweight Pattern
5. SDP; en verder?

DESIGN PATTERNS; WAT?

- 'Standaard' oplossing voor veelvoorkomende problemen
- Communicatie tool voor complexe design ideeën
- Helpt met leesbare, onderhoudbare code
- Separation of concerns



STRUCTURAL DESIGN PATTERNS; WAT?

- Oplossingen voor problemen met hoe klassen en objecten grotere structuren vormen
- Versimpelen van relaties
- Veelal; betere efficiëntie, flexibelere code
- Combineren van objecten voor nieuwe functionaliteit
- DRY principe

Een aantal hebben jullie al eens gezien;

- Adapter (Wrapper)
 - Zelfde gedrag, andere interface
 - S2; tekenen op verschillende displays
- Decorator
 - Zelfde interface, ander gedrag
 - S2; inverteren van een pin

SDP; PROXY PATTERN

Een placeholder object voor echte object om toegang tot het object te beperken of aan te passen.

Een aantal smaakjes:

- Remote Proxy
 - Handelt toegang tot remote objecten af
- Virtual Proxy
 - Crëert dure objecten on demand
- Protection Proxy
 - Controleert toegangsrechten tot objecten

Ook deze gebruiken jullie (hopelijk) stiekem al vaker: smart pointers



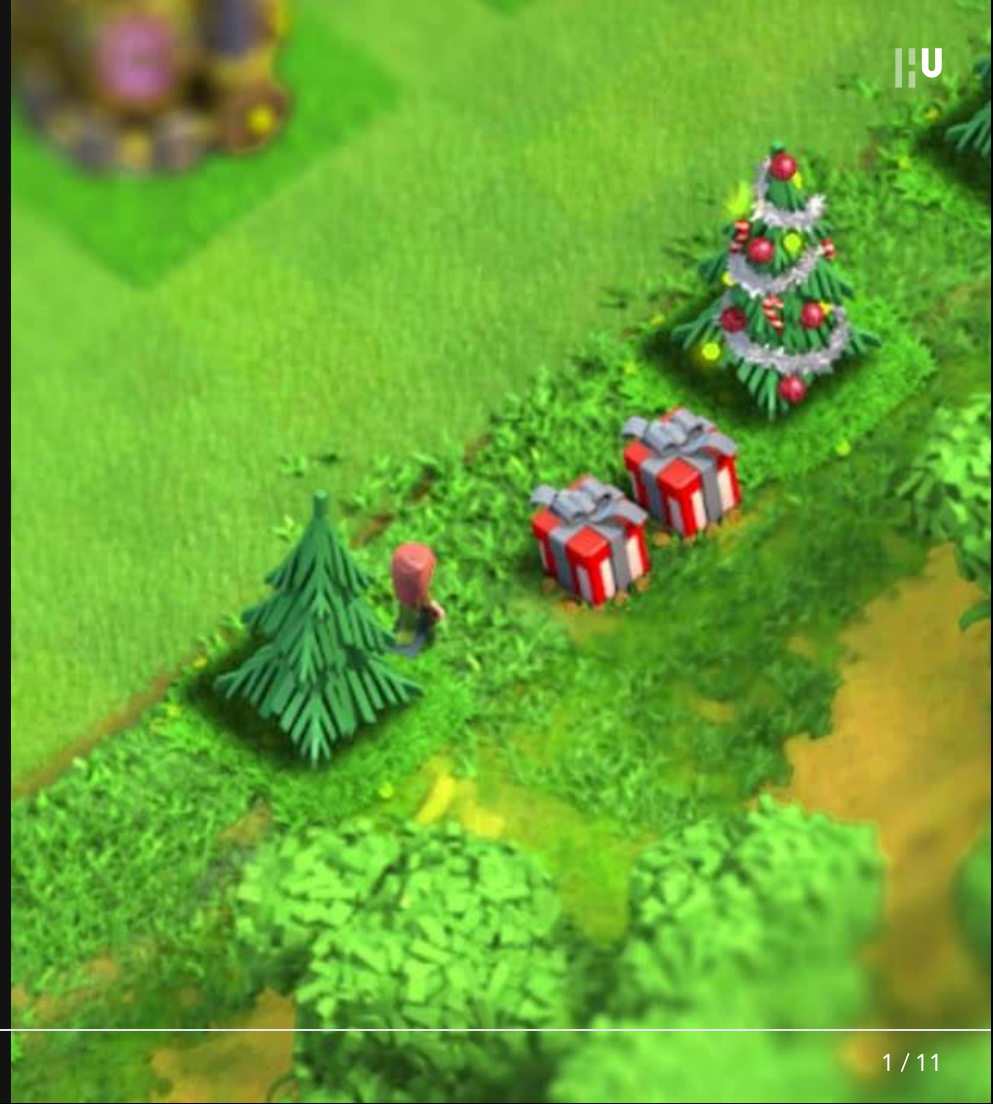


SDP; FLYWEIGHT PATTERN

- Voor verbeteren van performance
- Verminderen van geheugengebruik

Beide door het delen van gemeenschappelijke resources; géén kopieën dus!

- Intrinsieke vs Extrinsieke state
- Kan verder uitgebreid worden met een Flyweight Factory; voor later!





MESH

BARK

LEAVES

PARAMS
POSITION



MESH

BARK

LEAVES

PARAMS
POSITION



MESH

BARK

LEAVES

PARAMS
POSITION

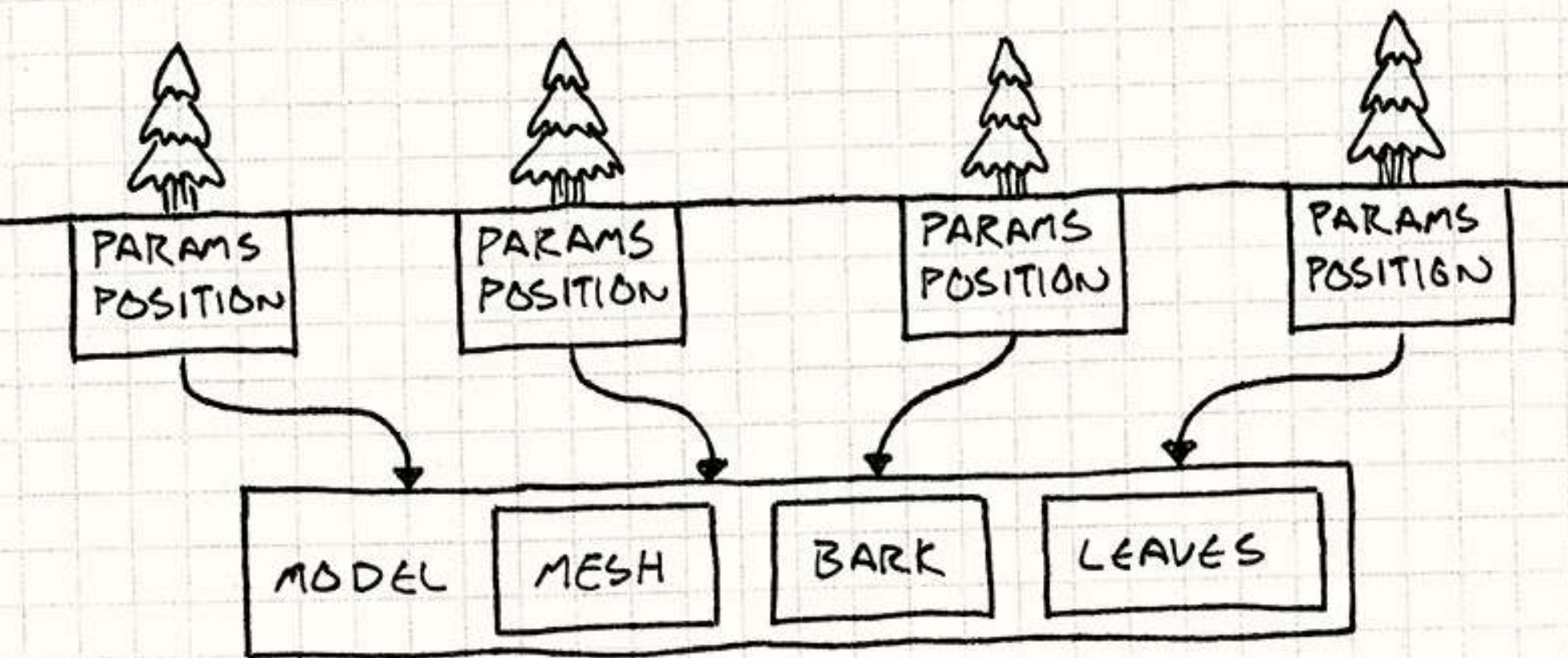


MESH

BARK

LEAVES

PARAMS
POSITION



SDP; EN VERDER?

Paar standaard patronen niet behandeld;

- Facade
- Bridge
- Composite

Meer over te vinden;

- <https://refactoring.guru/design-patterns/structural-patterns>
- <https://www.geeksforgeeks.org/structural-design-patterns/>

– E inde –