PROGRAMMING PATTERNS IN COMPUTER GAMES
CONTACT INFORMATION

Daniel Nael

daniel.nael@ut.ee

Daniel Nael#7435
PROGRAMMING PATTERNS
IN COMPUTER GAMES
SOFTWARE DESIGN PATTERNS

PROGRAMMING PATTERNS

IN COMPUTER GAMES
A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems.

It describes the problem, the solution, when to apply the solution, and its consequences. It also gives implementation hints and examples.

The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context.

-The “Gang of Four”
A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems.

It describes the problem, the solution, when to apply the solution, and its consequences. It also gives implementation hints and examples.

The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context.

-The “Gang of Four”
A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems. The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context.

- The “Gang of Four”
A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems.

It describes the problem, the solution, when to apply the solution, and its consequences. It also gives implementation hints and examples.

The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context.

- The “Gang of Four”
A DESIGN PATTERN ...

PROBLEM, SOLUTION, CONSEQUENCES

It describes the problem, the solution, when to apply the solution, and its consequences. It also gives implementation hints and examples.

The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context.

-The “Gang of Four”
A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems.

It describes the problem, the solution, when to apply the solution, and its consequences. It also gives implementation hints and examples.

The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context.

-The “Gang of Four”
A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems.

ARRANGEMENT OF OBJECTS AND CLASSES

The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context.

-The “Gang of Four”
A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems.

It describes the problem, the solution, when to apply the solution, and its consequences. It also gives implementation hints and examples.

The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context.

-The “Gang of Four”
GANG OF FOUR?

- Released in 1994
- First book on software design patterns
- Consists of 24 classic design patterns
- Examples in C++ and Smalltalk

None of this is relevant for the exam.
Book released by Robert Nystrom
http://gameprogrammingpatterns.com
Few copies also available in our library
SOFTWARE DESIGN PATTERN

- A recurring problem in object-oriented system
- The solution as general arrangement of objects
- Description of how to spot the problem
- Description of consequences of the solution
SOFTWARE DESIGN PATTERN

⇒ A recurring problem in object-oriented system
⇒ The solution as general arrangement of objects
⇒ Description of how to spot the problem
⇒ Description of consequences of the solution
A recurring problem in object-oriented systems

Description of consequences of the solution
SOFTWARE DESIGN PATTERN

- A recurring problem in object-oriented system
- The solution as general arrangement of objects
- Description of how to spot the problem
- Description of consequences of the solution
SOFTWARE DESIGN PATTERN

- A recurring problem in object-oriented system
- The solution as general arrangement of objects
SOFTWARE DESIGN PATTERN

- A recurring problem in object-oriented system
- The solution as general arrangement of objects
- Description of **how to spot the problem**
- Description of consequences of the solution
SOFTWARE DESIGN PATTERN

- A recurring problem in object-oriented system
- The solution as general arrangement of objects
- Description of how to spot the problem
- Description of consequences of the solution
General information: Courses web page
https://courses.cs.ut.ee/2022/programming-games

Study system: CGLearn web page
https://cglearn.eu
08.02: Introduction to Patterns in Game Development
15.02: Command
22.02: Command (continued)
01.03: Flyweight and Decorator
08.03: State
15.03: Strategy and Type Object
22.03: Observer
29.04: Singleton
01.04 - 04.04: Ludum Dare 50 (Optional bonus credits) | UT Mängujämm

05.04: Service Locator

12.04: Service Locator and Prototype

19.04: Essay Seminar

26.04: Factory

03.05: Object Pool

10.05: Builder

17.05: Exam consultations

24.05: Exam
Get help on or discuss previous task
Presentation on new topic and discussion
Work on the new task
Can be found at [CGLearn.eu > Tasks tab](https://cglearn.eu)

Tasks are also submitted at the same page.

Deadline is generally **2 weeks** from start of the task.

**Deadline runs out at 16:15.** Exactly at the start of a practice session.
There are a total of 100 points in the course.

- 70p CGLearn tasks
- 30p Exam

Grade is decided on the traditional grading scale
(51p E, 61p D, .., 91p A)
10p Researching a topic and presenting it in class.
(Available if you are struggling with points near end)

1p Exceptional or innovative solutions to tasks.
This, however, is rare.

5p Game jam participation. Date not yet announced.
QUESTIONS?