

# Software Engineering

## Modeling

### UML and Objects First

Lecture  
2009/10/02



# Content

- Modeling, an overview
- Concrete → abstract, scenarios and examples
- User Stories, and Usecases
- Object Diagrams
- Break
- Class Diagrams
- Sequence Diagrams
- Activity Diagrams



# Warm-up Questions

- Who of you likes programming (who not)?
- Who likes to program in “real” projects (who prefers just for fun)?
- Who uses modeling in their development environment (in “real” projects in fun projects)?
- Who likes to use modeling (private, work)?



# Why Model?

- *“If you were supposed to understand it, we wouldn't call it code.”* -  
from a Federal Express promotion, reported by IS Survivalist Matthew Persico
- Enables better communication
- Shows connections, relations, context at once
- Easier to see and estimate risks/costs



# Concrete, Abstract, Example

- Form teams of 3-4
- Discuss and write down examples for concrete and abstract things and for examples (total about 10 examples).
- Create definitions of concrete, abstract, and example
- Write down two ideas, what this could have to do with modeling.
- Work on this for 15 minutes.
- Report



# What is an example?

- Concrete!
- Wictionary (example):
  - *Something that is representative of all such things in a group.*
  - *Something that serves to illustrate or explain a rule.*
  - *Something that serves as a pattern of behavior to be imitated (a good example) or not to be imitated (a bad example).*
  - *A parallel or closely similar case, especially when serving as a precedent or **model**.*
  - *An instance (as a problem to be solved) serving to illustrate the rule or precept or to act as an exercise in the application of the rule.*



# Scenario (our definition)

- A story (should be very concrete, lots of details – not too many - assume the development project already exists - fiction)
- Long (half a page up to several pages)
- Describing the system to develop
- A case for things happening in and in interaction with the system



# User Story (our definition)

- Very concrete example of the problem to solve (assume the development project already exists)
- Very short (less than half a page) – fits on a flash card
- Write down as:
  - Title
  - Pre condition
  - Action (<3 sentences)
  - Post condition





# Scenarios and Examples

- Be specific and concrete not abstract
- “A user interacts with a system” is not an example
- “Klaus clicks on the login button and enters his username kla01 and the password klasecret into the name field of the appearing dialog” is an example (and a short scenario)



# Scenario: Interactive Towers of Hanoi



[http://en.wikipedia.org/wiki/Towers\\_of\\_hanoi](http://en.wikipedia.org/wiki/Towers_of_hanoi)

Discs should be moved interactively.



# How To Model Software?

- Use a standard
- UML
  - Unified Modeling Language
  - accepted Standard of Software Industry
  - managed by Object Management Group (OMG)
  - mainly used in OO-Design
- Stay critical
  - maybe other models more accurate than one of the UML



# UML – start for yourself!

- <http://www.uml.org/>
- <http://www.agilemodeling.com/essays/umlDiagrams.htm>
- [http://www.sparxsystems.com.au/UML\\_Tutorial.htm](http://www.sparxsystems.com.au/UML_Tutorial.htm)
- [http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)
- Estonian translation of Martin Fowler's UML Distilled:  
<http://www.cyber.ee/uml/>



# Free UML Modeler

- ArgoUML: <http://argouml.tigris.org/>
- Gaphor: <http://gaphor.sourceforge.net/>
- Umbrello: <http://uml.sourceforge.net/>
- Fujaba: <http://www.fujaba.de/>



# UML Overview

- **Behavior diagrams**

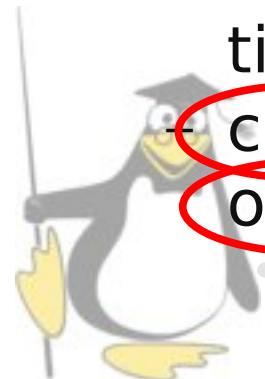
- behavioral features of a system or business process
- activity, state machine, use case, and all interaction diagrams

- **Interaction diagrams**

- subset of behavior diagrams emphasizing object interactions
- communication, interaction overview, sequence, and timing diagrams

- **Structure diagrams**

- depicts the elements of a specification irrespective of time
- class, composite structure, component, deployment, object, and package diagrams



# Used Material

- [http://www.sparxsystems.com.au/resources/uml2\\_tutorial/uml2\\_usecasediagram.html](http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_usecasediagram.html)
- [http://www.sparxsystems.com.au/resources/uml2\\_tutorial/uml2\\_objectdiagram.html](http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_objectdiagram.html)
- [http://www.sparxsystems.com.au/resources/uml2\\_tutorial/uml2\\_classdiagram.html](http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_classdiagram.html)
- [http://www.sparxsystems.com.au/resources/uml2\\_tutorial/uml2\\_activitydiagram.html](http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_activitydiagram.html)
- [http://www.sparxsystems.com.au/resources/uml2\\_tutorial/uml2\\_sequencediagram.html](http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_sequencediagram.html)



# Objects First Method

- Agile Software Development Method
- In close contact with customer
- Userstories and usecases
- -> Object and Usecase Diagrams
  - -> Class Diagrams
  - -> Activities (story diagrams)
- From concrete (specific) to abstract (generic).





# Objects First Method

- Scenarios
- User stories
  - Tests
  - base for measurement
- Use cases
- Object diagrams → Class diagrams
- Story diagrams
- What is different to your dev. process?



# Example User Story

- Name: Move red disk to second place
- Precondition: 3 towerplaces on table, red disc (size 1) on yellow (size 2) on blue (size 3) on orange (size 4) on initial towerplace (first)
- The player moves the red disc onto the second towerplace
- Postcondition: yellow, blue, and orange disc sorted on initial towerplace (of 3), red on second towerplace



# Create User Stories

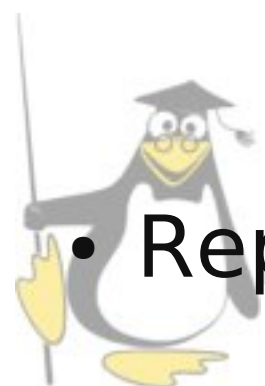
- Form teams (3-4)
- Create stories:
  - another positive concret move disc case
  - a failing move disc case (moving big disc on small disc)
  - Reset/Setup Towers of Hanoi
- Note down difficulties/problems
- Total time for activity: 20min
- Report!



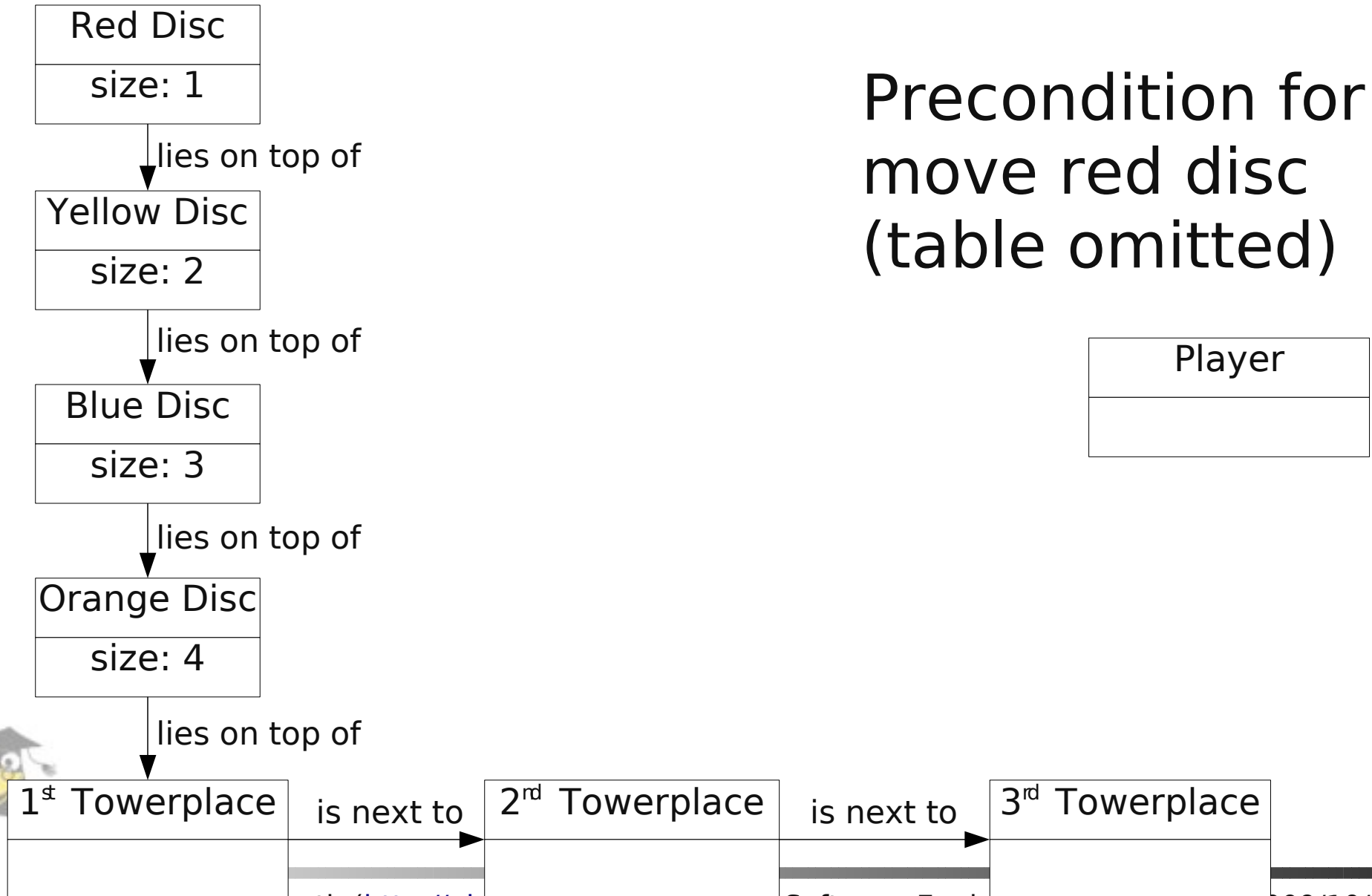
# Model an Object Diagram

- Team
- Object diagram for “Move red disk” and “reset/setup”, pre and post condition (25min)
  - pre- and post condition can be shown in one model (think how)
  - show change to from pre to post condition
  - what is special about this, what did you do for this change?
  - how can we go from here to class diagrams?

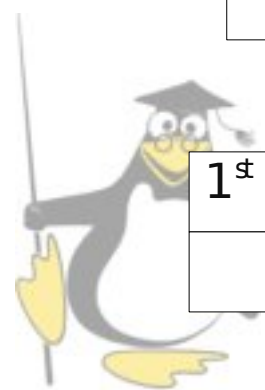
## • Report



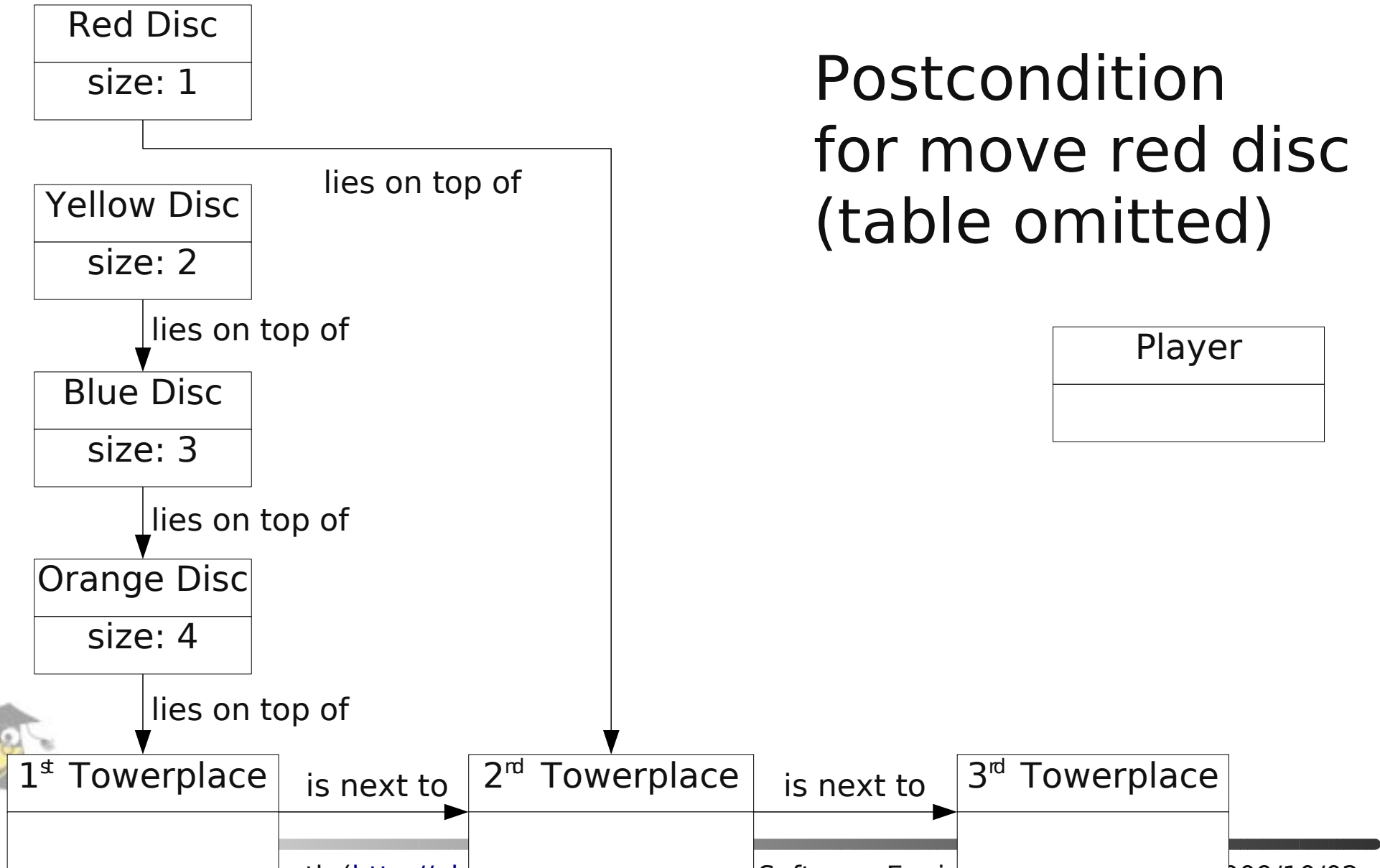
# Object Diagram



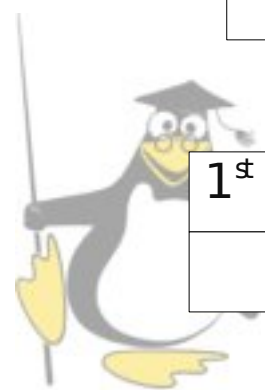
Precondition for  
move red disc  
(table omitted)



# Object Diagram



Postcondition  
for move red disc  
(table omitted)

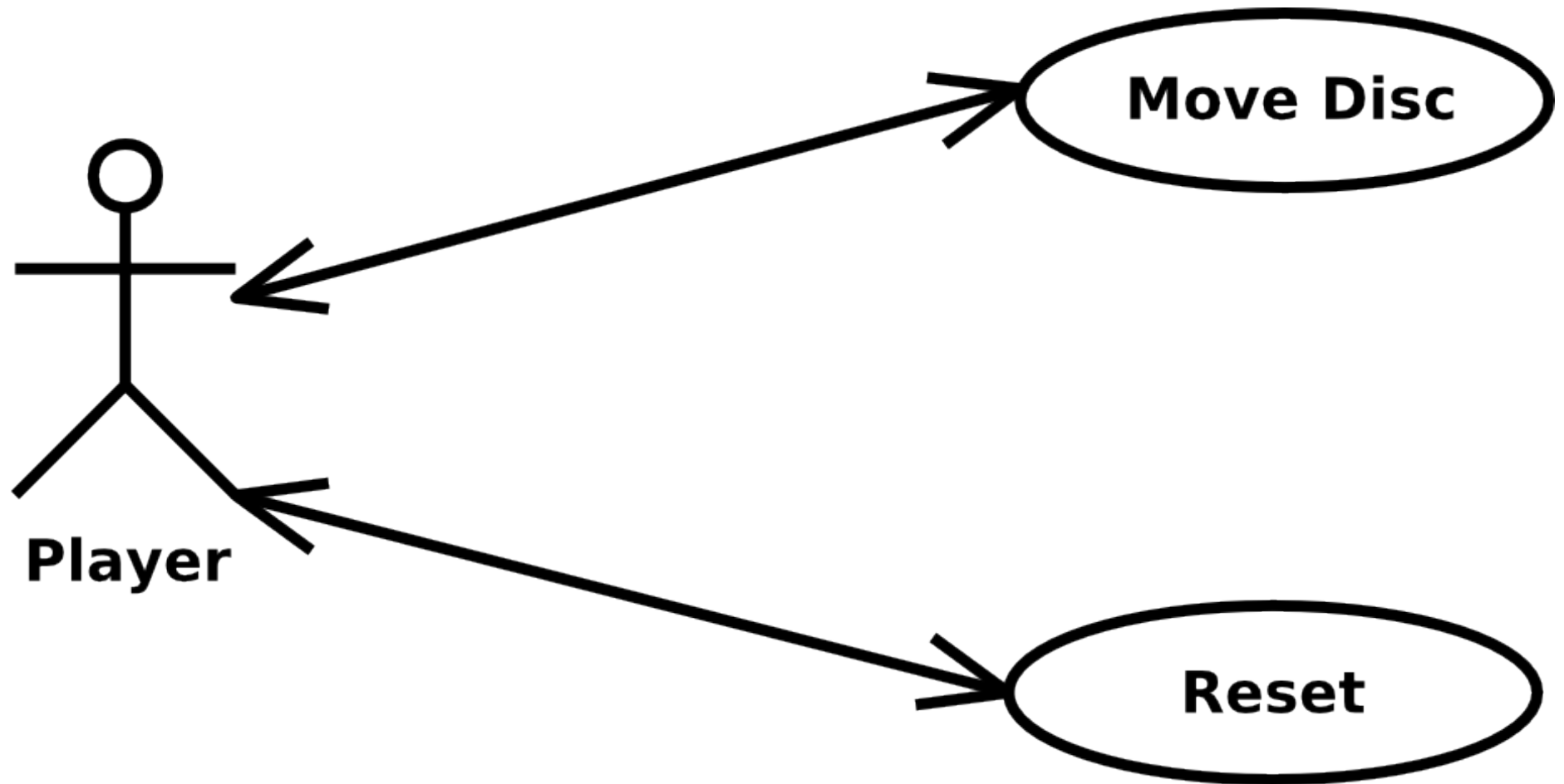


# Usecase

- Team up
- Create the two corresponding usecases
- Create usecase diagram
- Time: 15 min



# Usecase diagram





# Break

10 Minutes Break

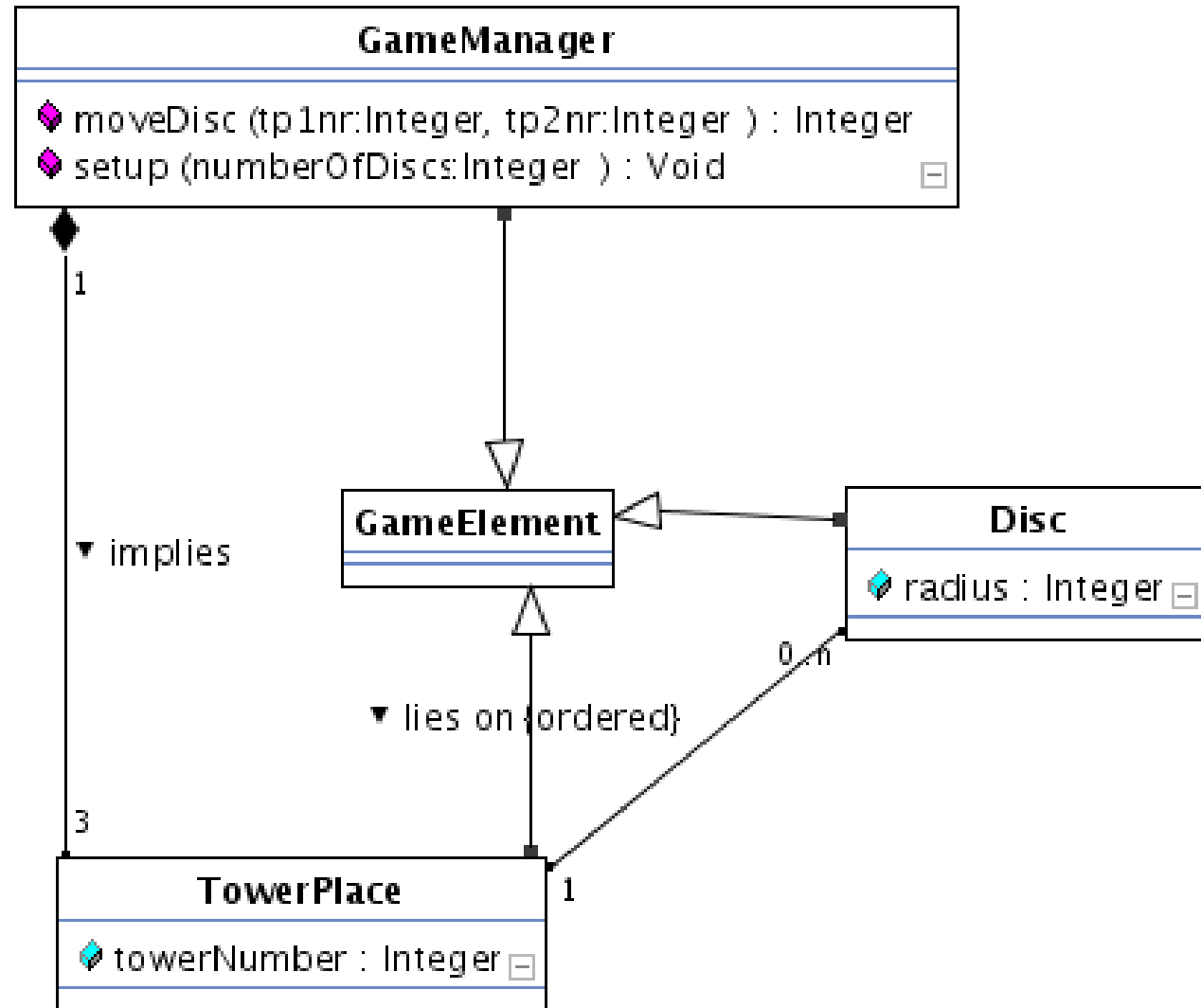


# Class Diagram Tasks

- Derive class diagrams from object diagrams (how will this be possible?)
- One team at blackboard
- Add attributes and methods (how to get methods?)
- Note down difficulties/problems
- Total time for activity: 30min



# Class Diagram



# Sequence Diagram Tasks

- Read description, read task
- One team at blackboard
- Create sequence diagrams
- Note down difficulties/problems
- Total time for activity: 30min



# Scenario (quite abstract)

- Customer brings items to the counter.
- Cashier scans each customer item.
- Cashier totals order, including tax.
- Cashier requests form of payment.
- Customer gives a credit card.
- Cashier scans card.
- Checkout system communicates scan data to verifier.
- Verifier reports that credit card payment is acceptable.
- Customer signs credit card slip.



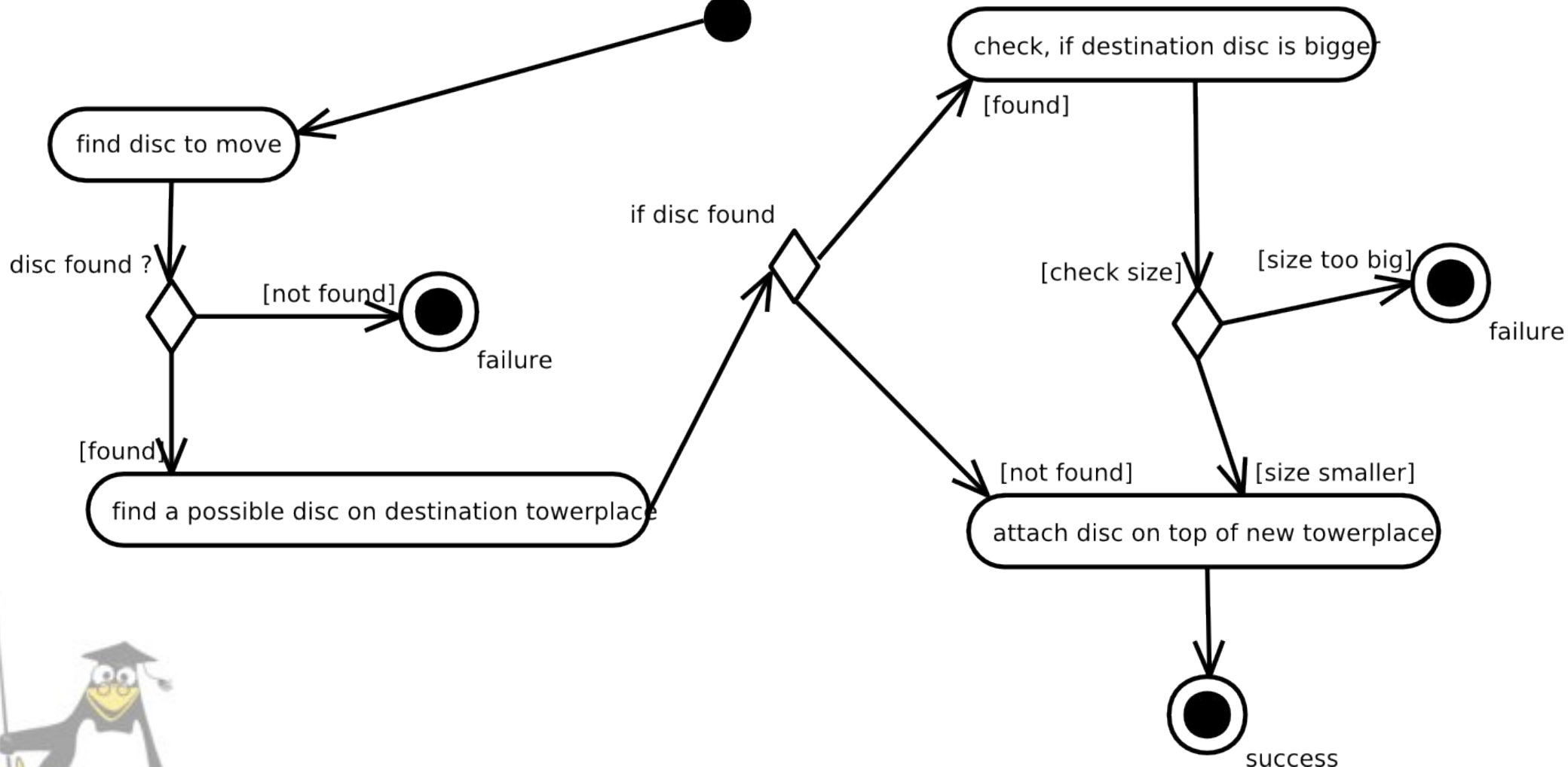
# Activity Diagram Tasks

- Read description
- One team at blackboard
- Create activity diagrams for move disc
- Note down difficulties/problems
- Total time for activity: 30min



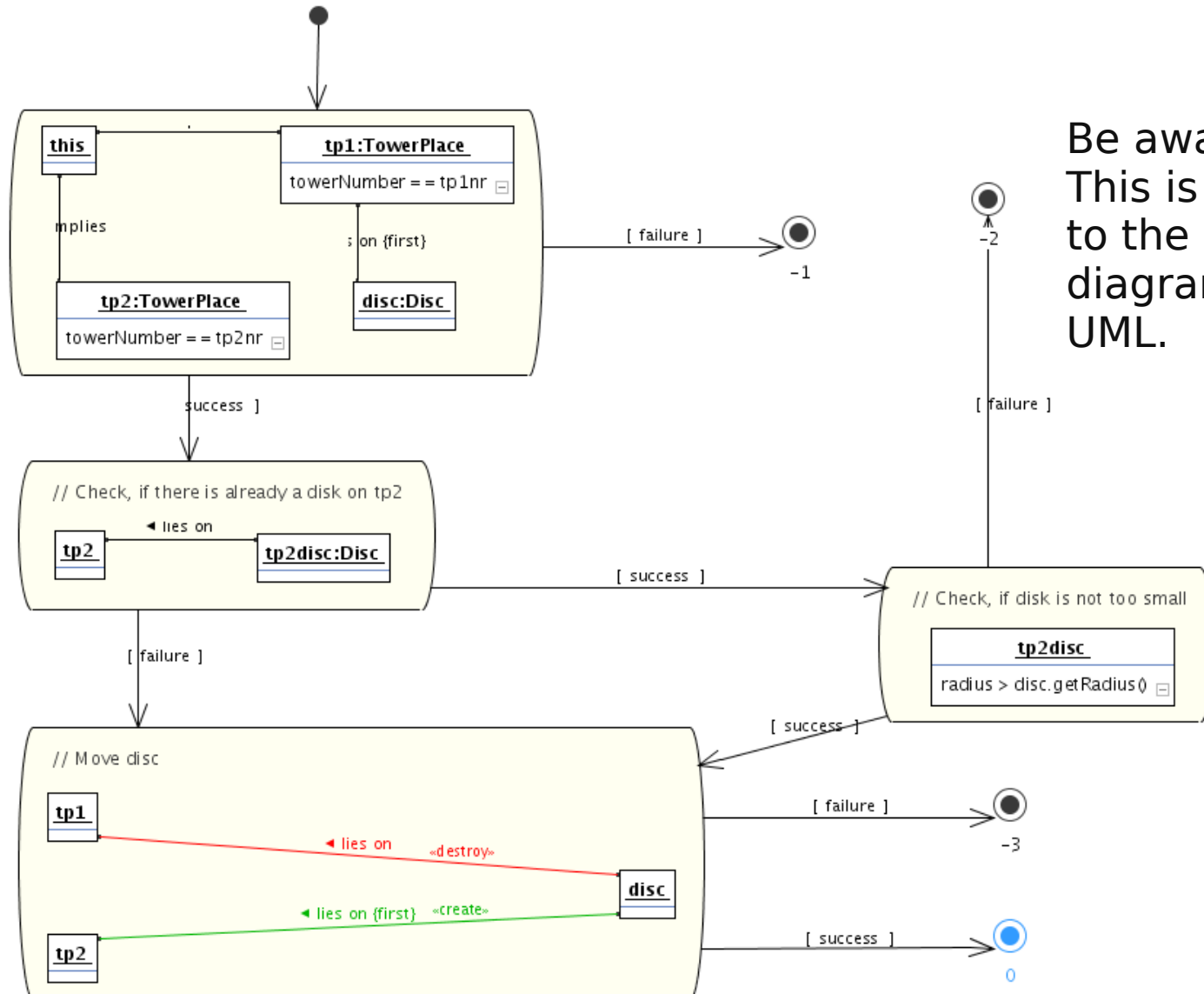
# Activity Diagram

moveDisc( towerplacefrom, towerplaceto )



# Story Diagram (Outlook)

GameManager::moveDisc (tp1nr: Integer, tp2nr: Integer): Integer



Be aware!  
This is an extension  
to the activity  
diagram defined in  
UML.





# Summary

- Why do we model?
- What is the Objects First Method?
- UML
  - user story, scenario
  - usecase
  - object diagram
  - class diagram
  - activity diagram
  - sequence diagram

