Design and Implementation [Iteration 1, Phase 3 & 4]

Slide Series 4

The Design Model

During these phases (3&4) we

- Construct an executable version of some core parts of the analysis model. This is v. 0.1 of the design model
- Get v. 0.1 up and running (during week 3)!

Common techniques

- Using UML sequence diagrams, upcoming ...
- Prototyping
- Hard coding, emulating missing parts, rudimentary design, normal flow, ... simplify!

...the above interact

- Diagram gives overview
- Prototyping clarify details, use in parallel

Starting Out

From RE and analysis in RAD we have

- A few high priority use cases
- The analysis model (class diagram with more or less "empty" classes and undirected associations)
- A GUI (mockup or possibly some basic implementation)

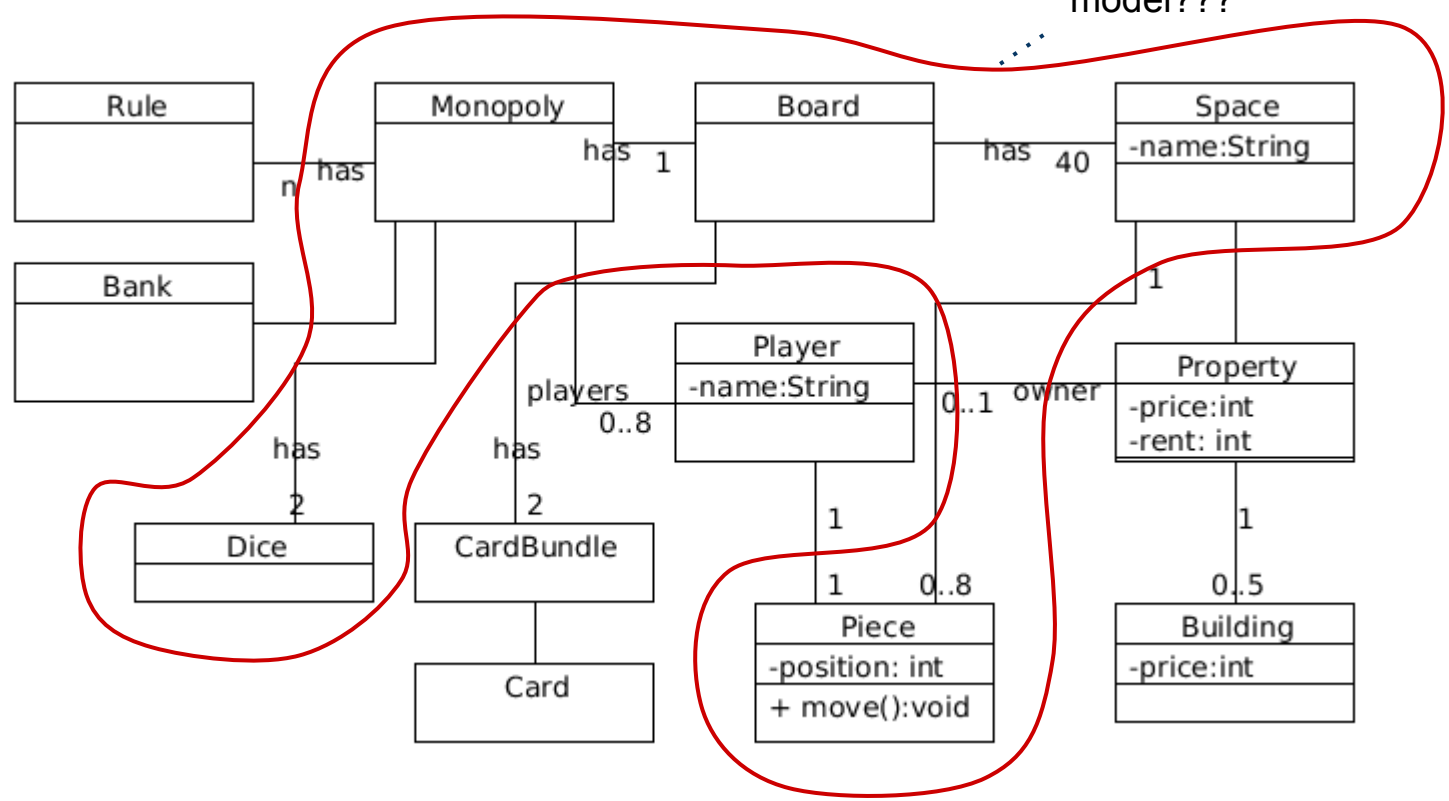
We pick 1-2 high priority use cases (and classes involved!)

Starting out with MP

Selected UCs: Move (first to be implemented) and EndTurn
Classes: .. at least Piece and Board, we'll see...

In first design model???

Analysis model from RAD (just a reminder)



Extending the Analysis Model

Add primitive data and "non-model" references to the involved classes (like Java API classes)

- Will distribute the basic responsibilities to objects (i.e. methods working on the data)
- Object having the data should act on the data (not pass around)

Primitive Data Player in MP

Example Player class primitive data

```
// Player
private final String name;  // Unique name for player
private boolean isBroke;
private int balance;
```

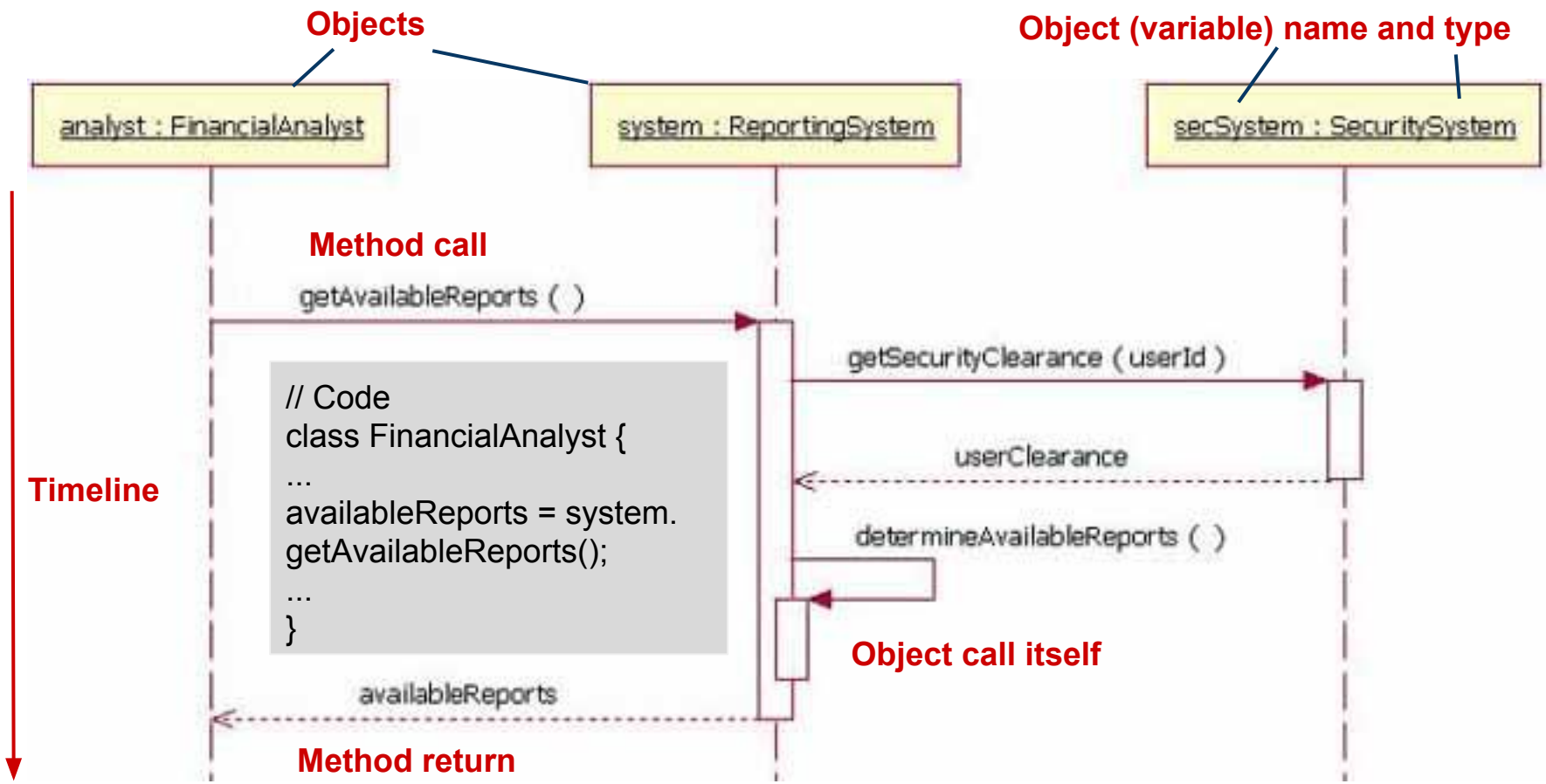
Other classes close to analysis model

Visualize Object Interaction

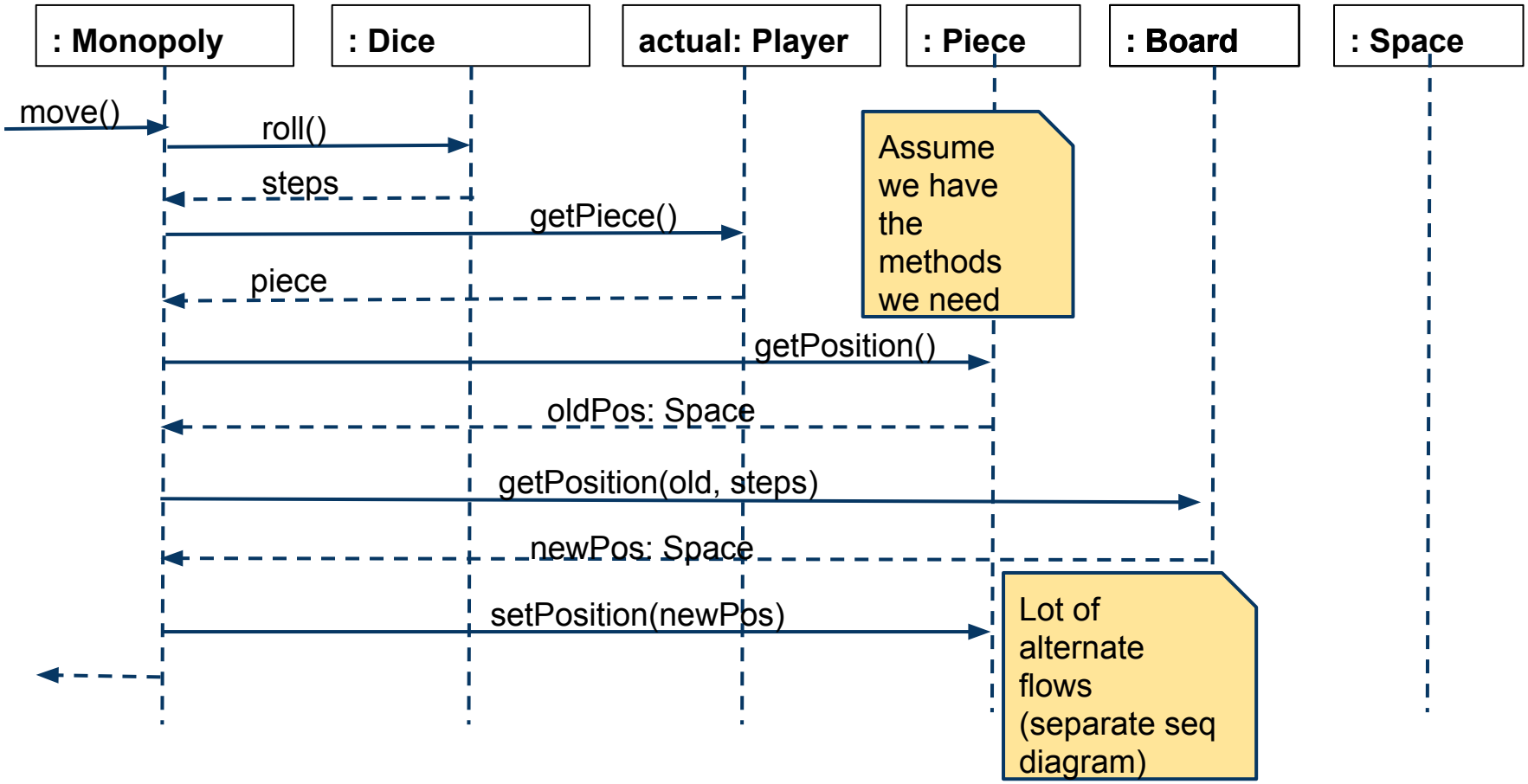
Next we start out with a sequence diagram

- Will create a call-chain for the UC
- Will give direction for association

UML Sequence Diagram



Sequence Diagram for UC Move from MP

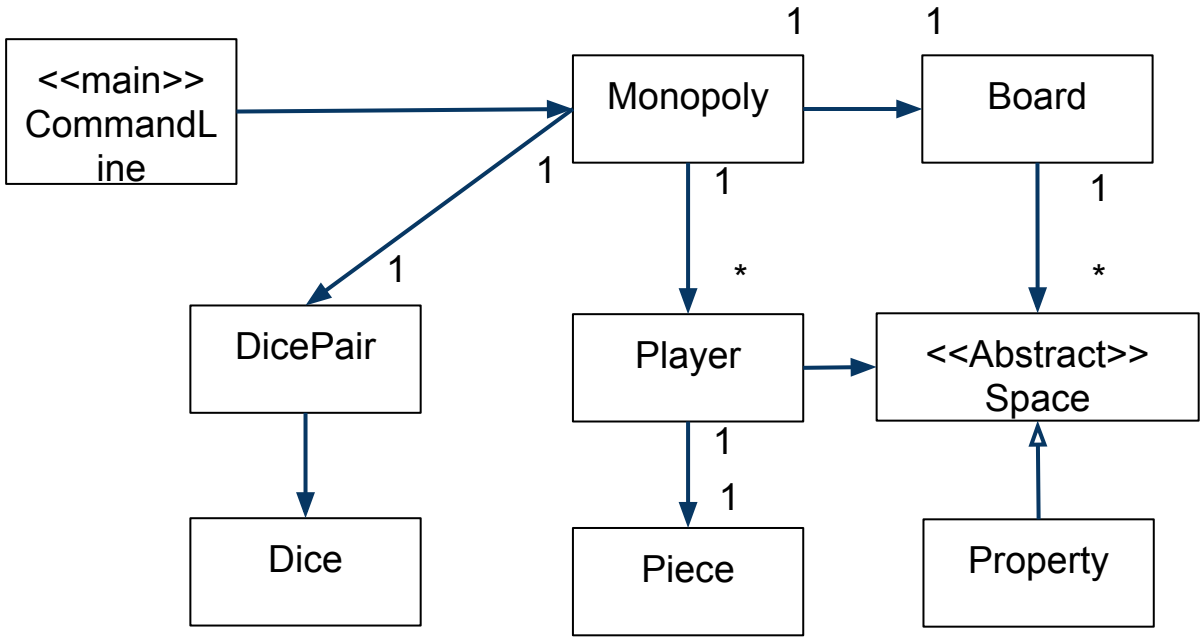


This is not the same as in code sample Monopoly v 0.1 (just an example)

Design Model v. 0.1 MP

Using analysis model, attributes, sequence diagrams and prototyping

- After some trial and error we get …



Technical helper classes

<<Static>>
Monopoly
factory

<<Static>>
Monopoly
options

This is a fat model (vs anemic). Data and logic in model
Anemic = just data + set/get, logic elsewhere

… the design model (v 0.1 runnable)

Implement UC Move MP

Should be fairly straightforward

- Create project
- Create classes
- Add constructors
- Add main method to a Main class (or a command line class upcoming)

If things feels awkward

- Swap direction of associations(?)
- Move data
- Missing classes?

Final Step to Run MP UC Move

Have GUI but simpler to run using a command line

- No obscuring GUI code
- No MVC for now, we call model directly

Create CommandLine class to run design model

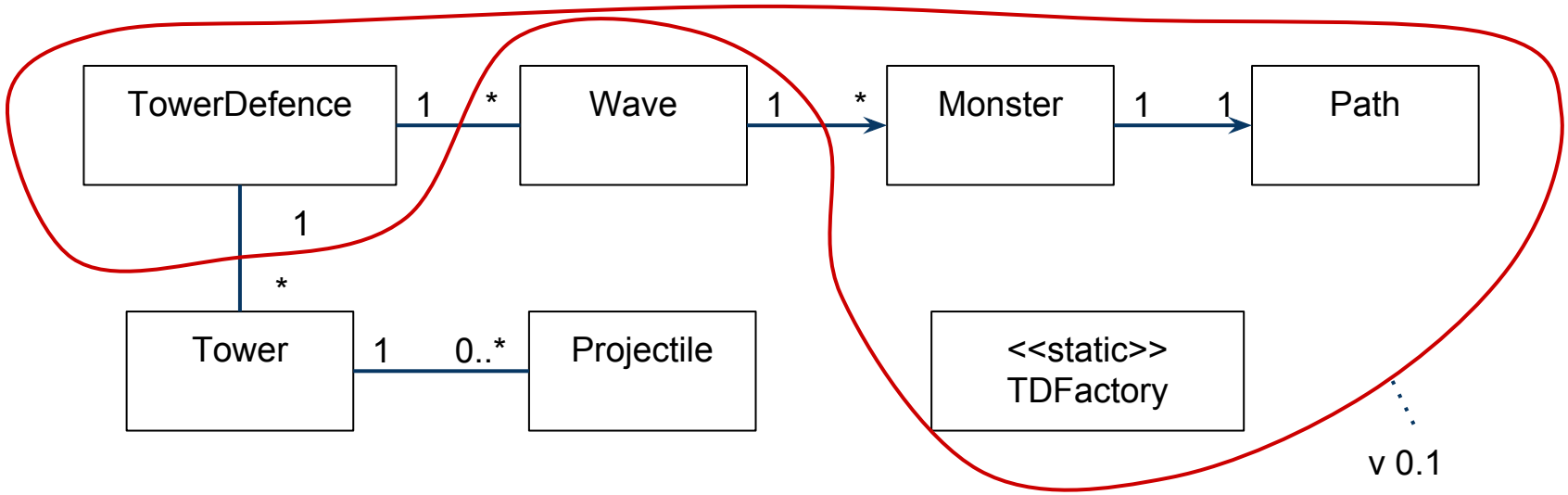
- Input: A Scanner in CommandLine class
- Output: override toString() and/or use System.out.println()
- Later we start more serious testing

DEMO time… MP version 0.1

Tower Defence Design Model

Application is very "graphical". How to run?

- Application advances by "tick" (discrete time units)
- Possibly try to run ticks manually (simple in Java 2D, a MouseListener, … other frameworks..????)
- Also: No MVC, do testing later



Summary

Using RAD as input

- We implemented a basic running version of our application
  - We selected a 1-2 high priority use case
  - Added attributes and basic methods to some classes
  - Created some UML sequence diagram from the use cases
  - From the above we got a very basic design model (= runnable analysis model) with attributes, directed associations and methods
- We did test runs of the model (UCs) from a simple command line
- No documentation for now (too early)

Next: Iteration 2