

---

# Exercise 4: The Settlers of Catan



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

## Software Engineering Design & Construction SS 2015 - Dr. Michael Eichberg

---

In this exercise, you will design a simplified version of the board game *The Settlers of Catan*<sup>1</sup>.

---

### Introduction

---

Make yourself familiar with the game rules as specified in [http://www.catan.com/files/downloads/catan\\_5th\\_ed\\_rules\\_eng\\_150303.pdf](http://www.catan.com/files/downloads/catan_5th_ed_rules_eng_150303.pdf). We assume a reduced set of game components, i.e., **there are no**

- sea frame pieces,
- harbor pieces
- Development Cards,
- Special Longest Road or Largest Army Cards,
- and no robber.

You can ignore all rules that refer to any of these cards or pieces. For example, nothing happens if a player rolls a 7. Your task is to design *The Settlers of Catan* as a computer game. Adhere to the five SOLID principles (Single Responsibility, Open Closed, Liskov Substitution, Interface Segregation and Dependency Inversion).

First, think about a design for your class hierarchy. Decide for which game components you need classes and which can be omitted or represented differently for a computer simulation. There should be two kinds of players, human and computer players.

Second, think about how a turn of a game of two proceeds. Imagine it is the turn of a computer player in which it should

- attempt to trade a non-zero amount of resources with the other player (you choose the exact trade), and
- build a road and a settlement if it has the required resources<sup>2</sup>.

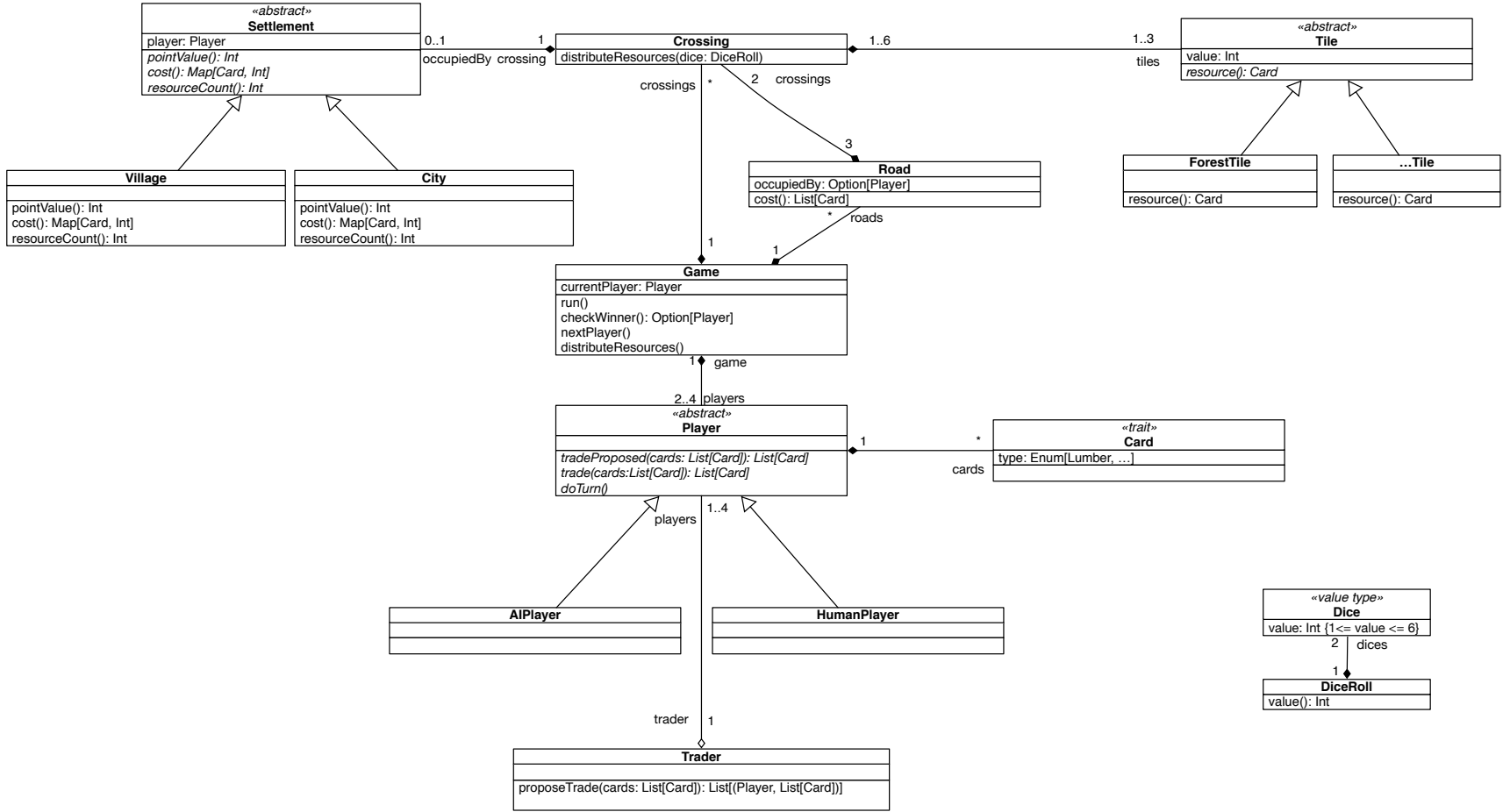
Think about the interface between a player and the game and how they interact with each other. Note that the game logic needs to prevent cheating (doing things in the wrong order, building things the player doesn't have the resources for etc). Reduce the number of ways a player can cheat by carefully designing the interface and interactions between the game logic and the player.

Since designing software is an incremental process, you might need to adapt your class design to address any deficiencies you discover while thinking about the turn design.

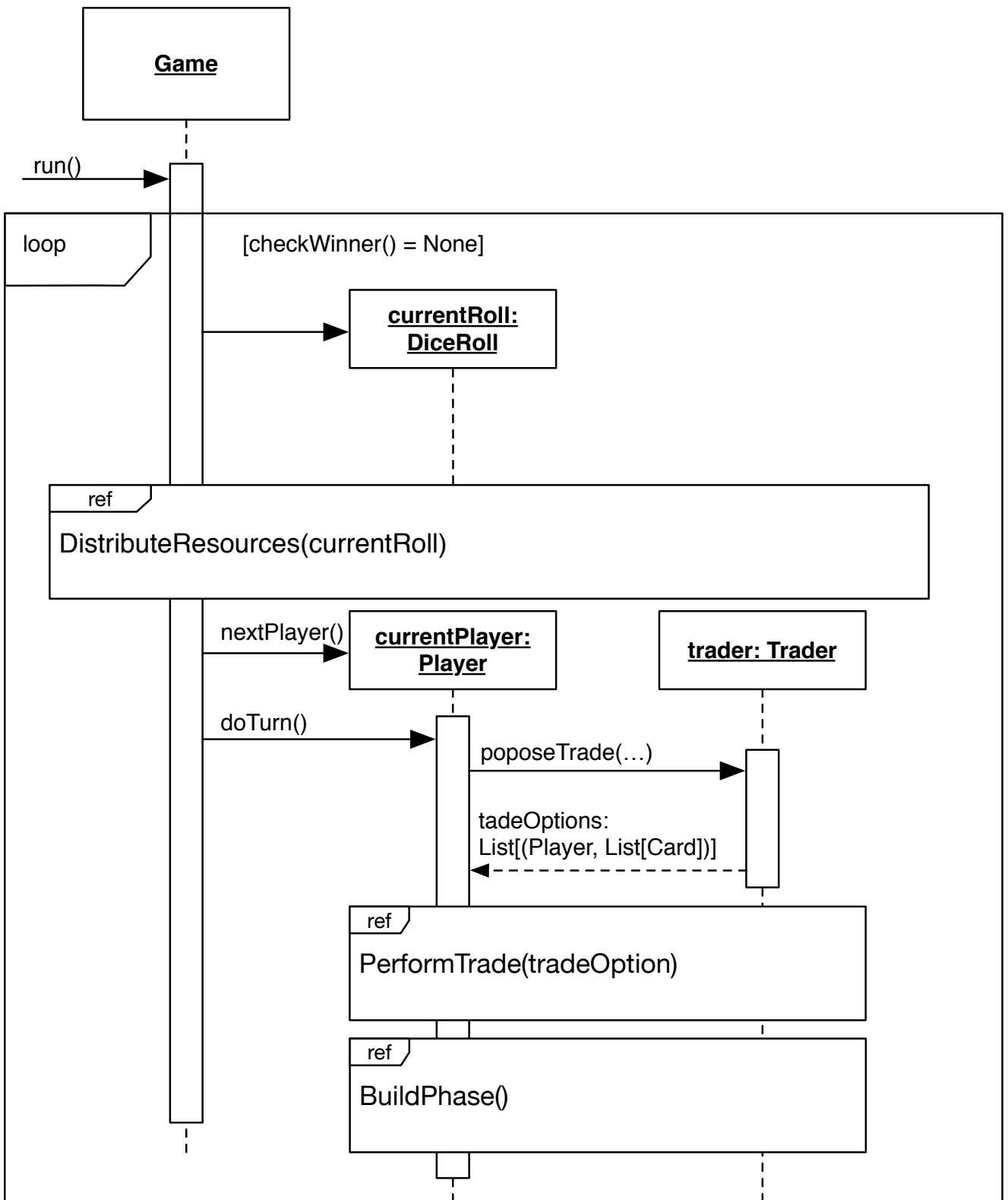
---

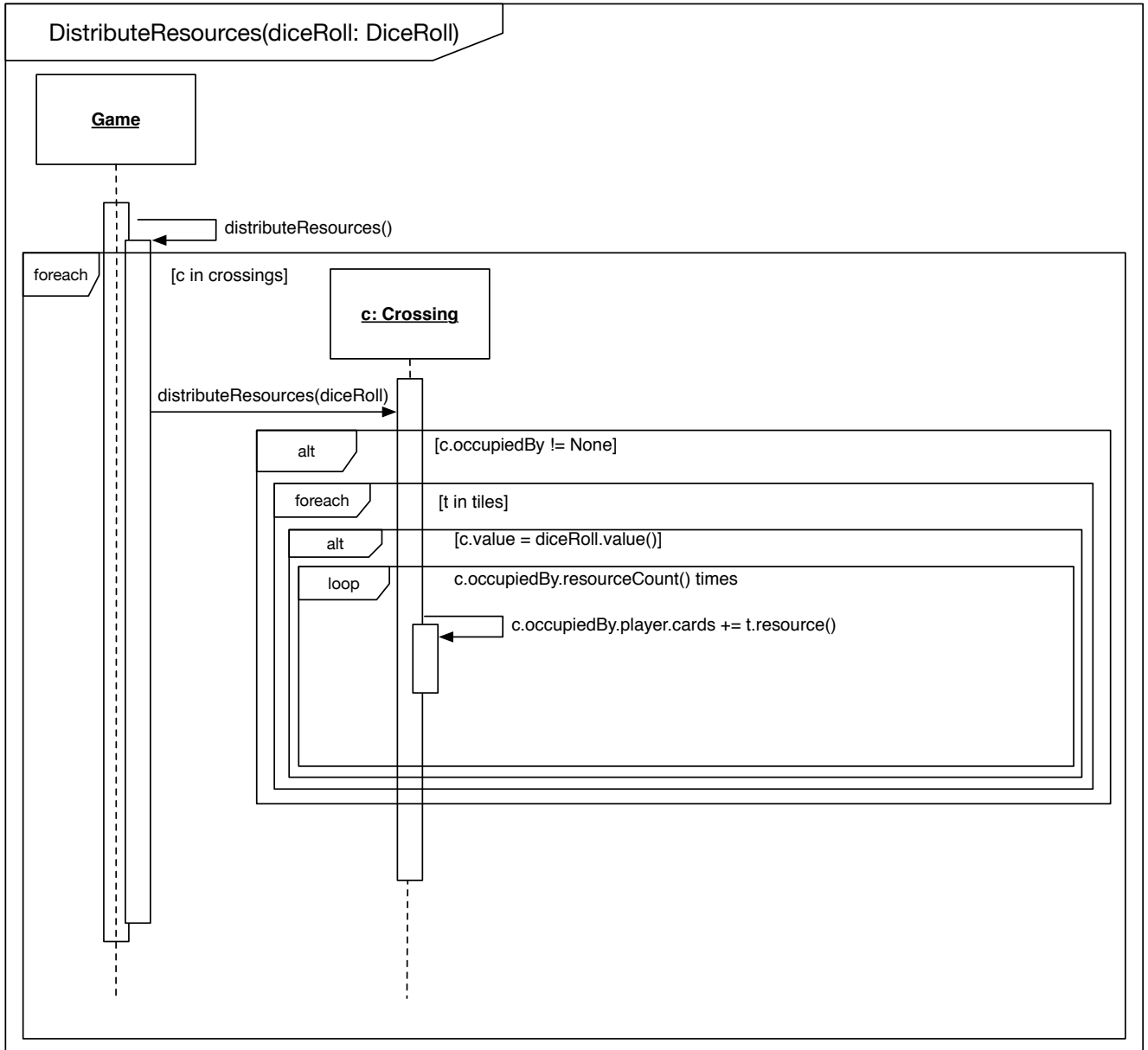
<sup>1</sup> <http://www.catan.com/>

<sup>2</sup> You might want to read <http://www.ibm.com/developerworks/rational/library/3101.html> for how to model guards and conditions in a sequence diagram.



Task 2 Turn design





### Task 3 SOLID

#### Task 3.1 Responsibilities

- **Dice:** Represents a single dice (a value between 1 and 6). Produces random values in that range.
- **DiceRoll:** Represents a dice roll in settlers. It consists of two dice. Produces a fair two dice roll.
- **Game:** Contains the main game logic and the board. Manages turns.
- **Tile:** A single board tile. Has a dice value (pays out on that value) and an associated resource.
- **Crossing:** Connects up to three tiles and roads. Can be occupied by a settlement.
- **Road:** A road between two crossings. Can be occupied by a player.
- **Settlement:** A settlement belongs to a player and is either a village or a city.
- **Village:** A settlement that produces one resource.
- **City:** A settlement that produces two resources.

- 
- **Player:** Players can be either human players or AI players. Players are responsible for handling turns (trading, building...).
  - **Trader:** Manages trading between players.
  - **Card:** Awarded for villages and cities producing resources. Used for building.

---

### Task 3.2 S.O.L.I.D

---

- **SRP:** The design honours the single responsibility principle as the classes are kept small. E.g. the player class consists only of methods relating to a turn and the therefore needed methods.
- **OCP:** The design honours the OCP principle. It is open for extension on both Tiles and settlements.
- **LSP:** LSP holds for the inheritance relations in the class diagram.
- **ISP:** There are no clients of any interface / trait in our design that are forced to depend on methods they do not use, so we adhere to ISP
- **DIP:** We adhere to DIP as for example Game does not depend on HumanPlayer, but only on Player which is a abstract class which all kinds of players have to inherit from. The same holds for Settlements and Tiles.