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*[1].

Introduction

Make yourself familiar with the game rules as specified in `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 theses 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[2].

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.

Task 1 Class design

Draw a UML 2.0 class diagram of your class hierarchy. Give inheritance relationships and associations (do not forget to provide correct multiplicities!). Also provide methods you need to simulate a turn.

Task 2 Turn design

Draw a UML 2.0 sequence diagram for the sample turn above in order to demonstrate the interactions between objects of the different classes. For simplicity reasons, assume that all checks against cheating pass. Your diagram should focus on the interactions between the game and a player. For example, it is not important to go into detail *how* the player finds the position of where to build a road, but it needs to get a handle to that position from some method call (model the AI as a black box). There should be no free variable names in your diagram (cfg. the slides for the first exercise)!

---

[1] `http://www.catan.com/`
[2] 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 3 SOLID

a) Shortly describe the responsibilities of each class.

b) Write for each SOLID principle, how your design honors it. For the Open Closed Principle, name two extensions your design makes possible.

Solve it on your own!

Although this exercise is not graded, it is highly recommended to do it by yourself. Just looking at a solution is much easier in comparison to actually coming up with it.

Requirements if you want to submit your solution

You can, once in the semester, submit your solution to get it corrected. Send your solution to weiel@st.informatik.tu-darmstadt.de. Make sure you submit exactly one pdf file that contains your solution. In this exercise there is no code to be submitted!