Software Specification and Design - Week 1

By Keeratipong Ukachoke

About me

- Keeratipong Ukachoke (You can call me กี)
- Kasetsart University - 2011
- Johns Hopkins University - 2013
  (Fulbright Scholarship)
- Cofound Fameworks - 2014
  (fameworks.co)

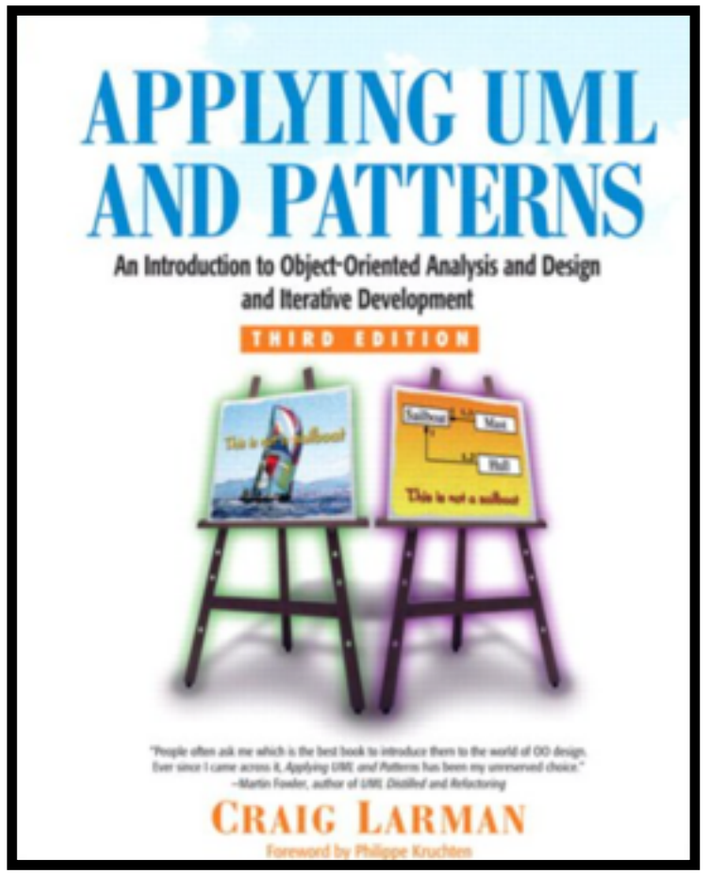
Contact

- Facebook: Keeratipong Ukachoke
- kee@fameworks.co
- 091-880-0902

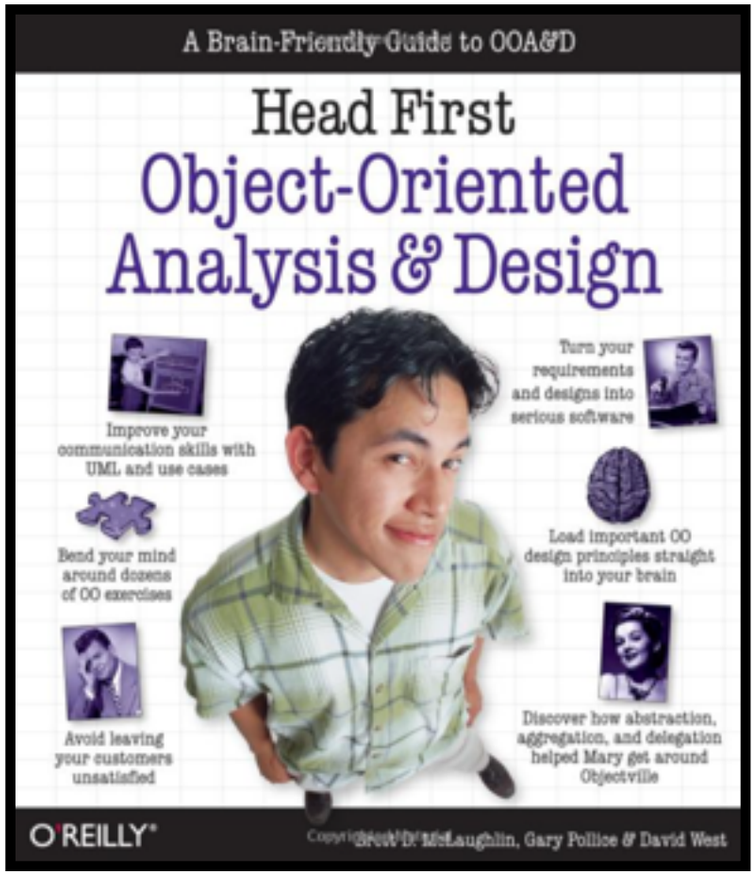
What you will learn

- **OOA/D**
- A bit about processes
- Requirements analysis
- Applying UML
- Design Patterns & Design Principles
- Interesting technical knowledge

Class Materials



ISBN: 0131489062



ISBN: 0596008678

Class Structure

- 1 hour 45 minutes - Following the book(s)
- 15 minutes break
- 60 minutes - More technical stuffs

Mutual agreements

- Anything I teach in the class can be in the exam
  (Sometimes they are not in the book)
- I will not talk on every details
- It’s your duty to read on the topics
- Learning by doing

Grading

- 35% Midterm
- 35% Final
- 20% Assignments & Quiz
- 10% Participation
  (Start from 5%, 0.5% - 1% per answer)

A	: 85% - 100%
B+	: 80% - 84%
B	: 75% - 79%
C+	: 70% - 74%
C	: 65% - 70%
D+	: 60% - 64%
D	: 55% - 59%

Analysis vs Design

- Analysis - investigation on the problem
- Design - find conceptual solutions

OOA/D?

- Object-oriented analysis - finding and describing the **objects** from the requirements

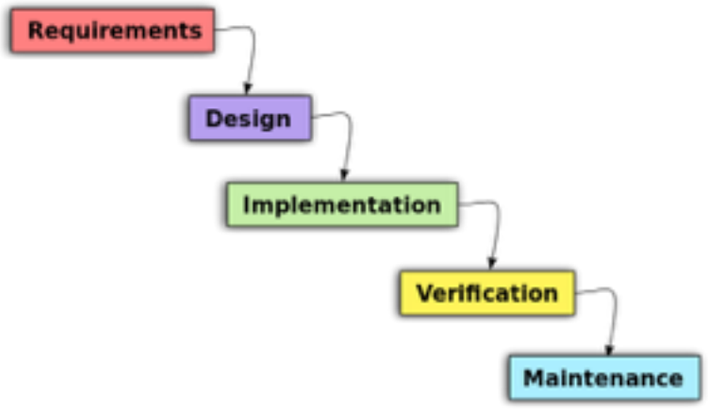
- Object-oriented design - defining **software objects** to fulfil the requirements

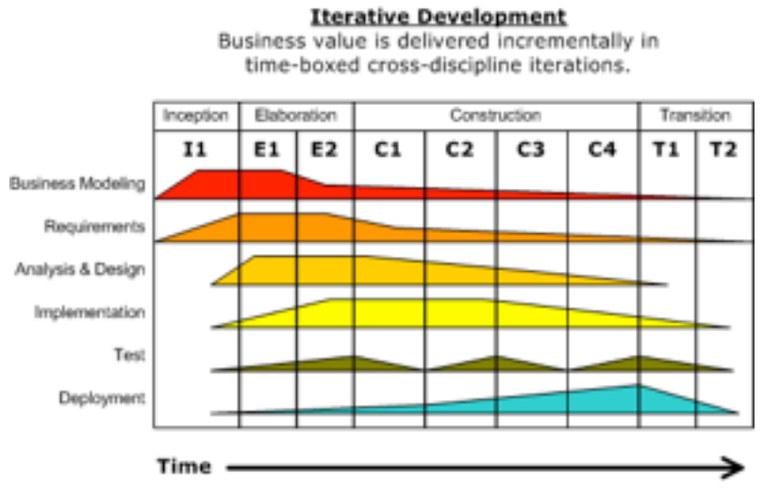
OOA/D

OOAD is a popular technical approach to analyzing, designing an application by applying the object-oriented paradigm and visual modeling throughout the development life cycles to foster better stakeholder communication and product quality.

OOA/D

- It’s one of many technical approaches
- Can be applied to any processes





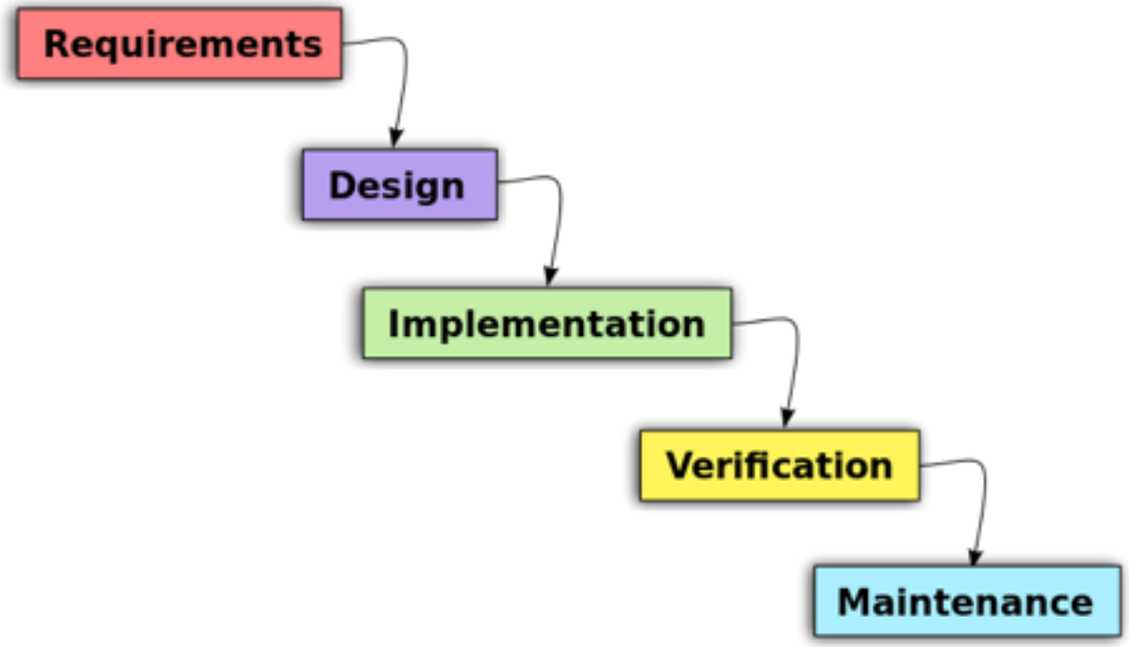
Source: Wikipedia

Dice game

- Super simple example
- ‘Dice game’
  - Player roll two dice
  - Win if the sum is 7
  - Otherwise, lose

Dice game

Let’s us follow the waterfall model



Source: Wikipedia

Dice game - OOA/D steps

- Define use cases (Requirements)
- Define domain model (Requirements & design)
- Define interaction diagram (design)
- Define class diagram (design)
- Write some codes together!
- Write tests

Dice game - use cases

- Part of requirements analysis
- Written stories
- Dice game's use case - brief version
    - Use case [**Play a dice game]**
        1. A player requests to roll two dice.
        2. System presents results.
        3. If the sum of faces is 7, player wins, otherwise, player loses.
- More about it in later chapter

Dice game - domain model

- Part of object oriented analysis
- Noteworthy domain concepts or objects
- Can be expressed as a diagram

Dice game - domain model (2)

- Let’s look back at the use case
- Dice game’s use case - brief version
    - **Play a dice game** - A player requests to roll two dice. System presents results. If the sum of faces is 7, player wins, otherwise, player loses.

Dice game - domain model (2)

- Let’s look back at the use case
- Dice game’s use case - brief version
    - **Play a dice game** - **A player** requests to roll two **dice**. **System** presents results. If the sum of faces is 7, player wins, otherwise, player loses.

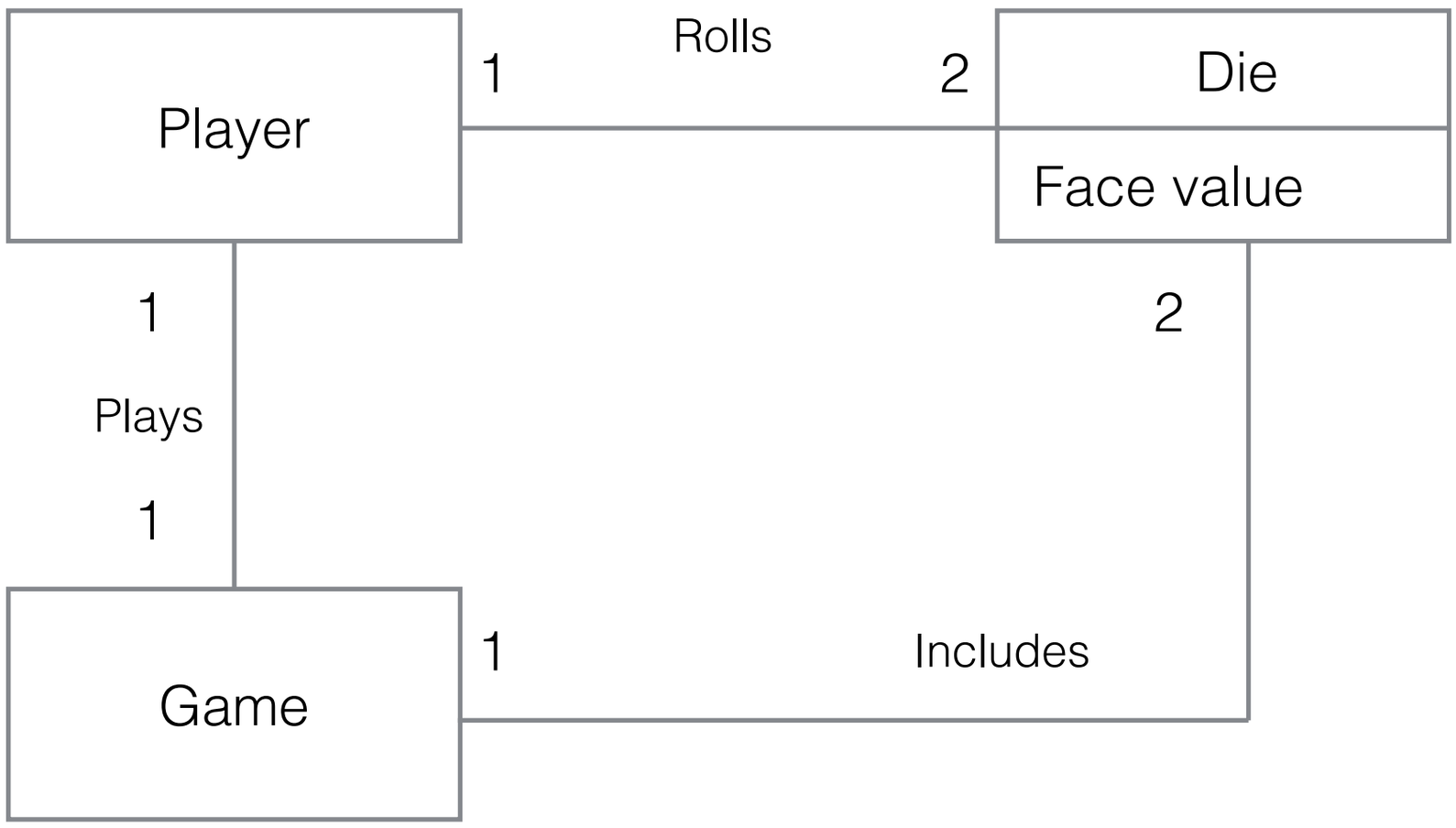
Dice game - domain model (3)

Player

Die

Game

Dice game - domain model (3)



Dice game - domain model (4)

- DOMAIN MODELS ARE NOT SOFTWARE OBJECTS

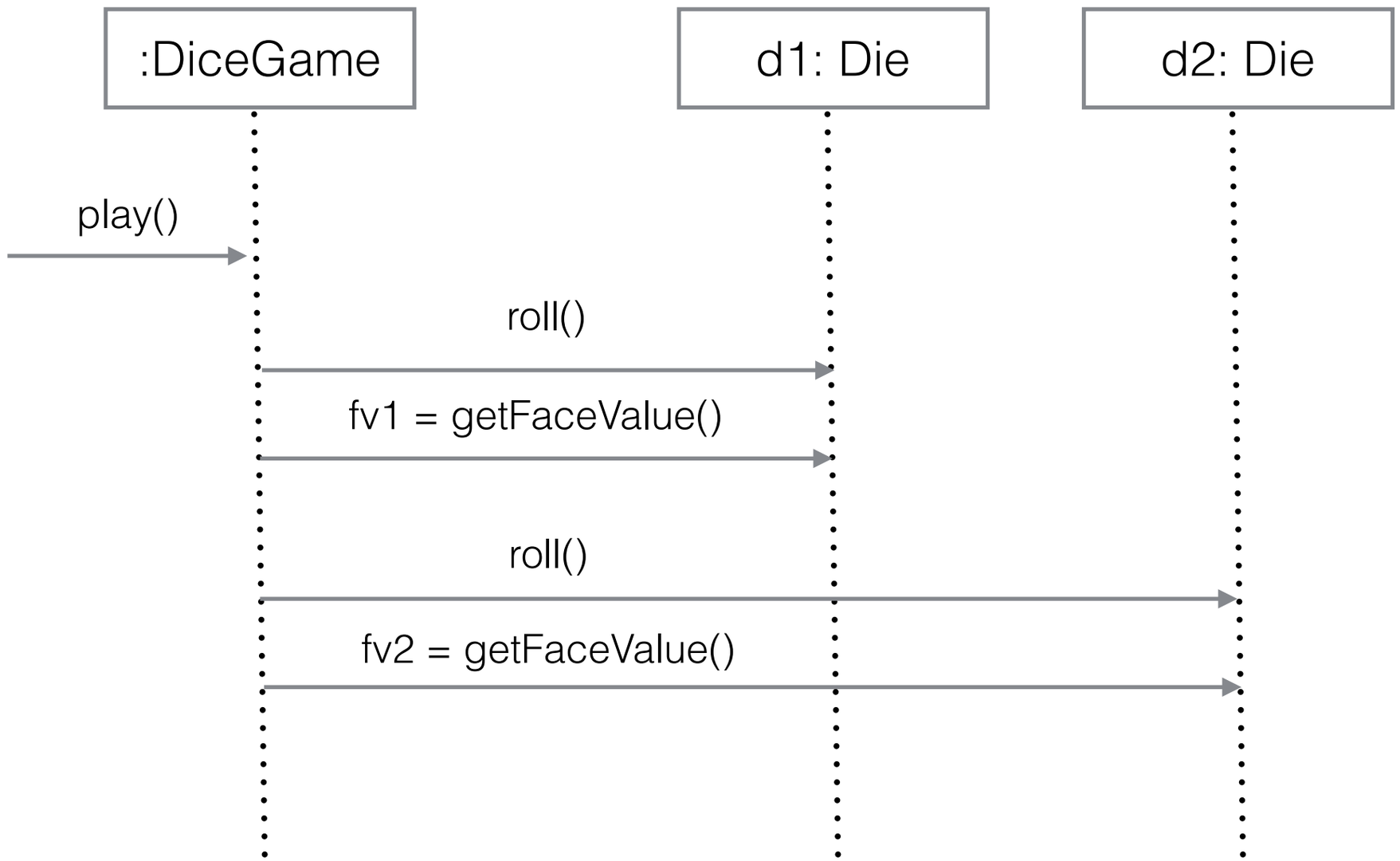
- However, they are related. We might have a class called Die later.

- We will cover more details later

Dice game - interaction diagrams

- Part of object oriented design
- Dynamic
- Example, sequence diagram
- Let’s see an example

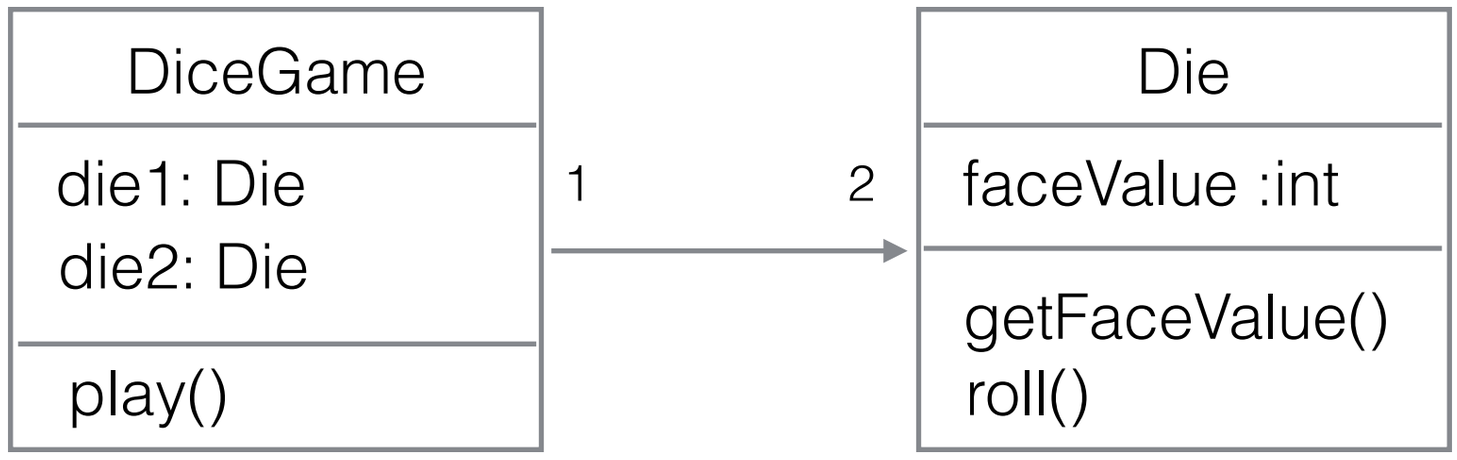
Dice game - interaction diagrams (simple version)



Dice game - class diagrams

- Part of object oriented design
- Static
- Attributes & methods
- Let’s see an example

Dice game - class diagrams (2)



Let’s write some code

- Using what we already have
    - Use cases
    - Domain models
    - Class diagrams
    - Sequence diagrams

Interaction diagram vs Class diagram

Benefits of Interaction diagram vs class diagram

Some real life example.

Let’s take a look at my current project.

Card Game example

- Develop a card game that let 2 players draw 2 cards for each of them from a shuffled deck, player with the bigger (sum%10) win. If the sum are equal, then draw. J,Q,K are treated as 0. A is 1. There is no joker in the deck.

Card Game example

- Develop a card game that let 2 players draw 2 cards for each of them from a shuffled deck, player with the bigger (sum%10) win. If the sum are equal, then draw. J,Q,K are treated as 0. A is 1. There is no joker in the deck.

Card Game - Use case

- Use case (simple version)
    - Shuffle the deck
    - First player draws the first card
    - Second player draws the first card
    - First player draws the second card
    - Second player draws the second card
    - Both of them show the cards
    - Player with the higher (sum % 10) wins.

Card Game - Domain model